UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Abdul Wahab

Hackathon Organiser Awareness Dashboard

Master's Thesis (30 ECTS)

Supervisor: Alexander Nolte, PhD

Tartu 2021

Hackathon Organiser Awareness Dashboard

Abstract:
There is little support available for organisers to keep an overview of the online hackathon they are running. Support for organisers of colocated hackathons is well-established in literature and practise,yet, it does not cater to the challenges posed by online hackathons. This study aims to determine how existing infrastructure can be used to provide overview of online hackathons to organisers. Specifically, it investigates what information is required to provide hackathon organisers with the overview, how to obtain this information and present it to hackathon organisers. In this context, we explore the concept of awareness from CSCW literature to develop a theoretical foundation for information required to provide an overview to hackathon organisers. We use raw chat data gathered from four hackathons to extract this information. We design a dashboard styled web application to present this information to hackathon organisers.

To validate the relevance of information and to evaluate the usability of the dashboard. The participants were presented with scenarios that hackathon organisers can typically face in an online hackathon. Their interaction with the dashboard was recorded and we followed up with questions. The data collected from study was then analysed to validate the relevance of information presented and further identify means of improvement of our dashboard. Although participants approve of the overall dashboard design, improvements to the dashboard design are structured according to their importance. The results suggest that hackathon organisers find the information presented on the dashboard to be helpful in keeping an overview of the hackathon.

Keywords: Hackathon, Awareness, Dashboard, Organiser
CERCS: P170 - Computer Science, numerical analysis, systems, control

Hackathoni korraldaja teadlikkus armatuurlaud

Lühikokkuvõte:
Praegu puudub häkatoni korraldajatel piisav võimekus haldamaks veebis korraldatavaid häkatone. Traditsiooniliste häkatonide puhul, kus inimesed viibivad füüsiliselt koos, on välja kujunenud tavad ning leidub ka kirjandust, mis toetab häkatonide korraldamist, kuid neid eksisteerivaid võtteid ei saa üle kanda veebis toimuvatele üritustele. Käesolev uurimus proovib selgitada, kuidas saab kasutada eksisteerivat taristut, et anda korraldajatele parem ülevaade veebipõhisel häkatonil toimuvast. Täpsemalt keskendub uurimus sellele, millist informatsiooni häkatoni korraldajad vajavad üritusest parema ülevaate saamiseks, kuidas seda informatsiooni saada ja kuidas seda häkatoni korraldajatele esitleda. Uurimuse raames me tutvume CSCW kirjandusega, et omandada teoreetiline arusaam sellest, millist informatsiooni on tarvis, et häkatoni korraldajad saaksid üritusest parema ülevaate. Vastava info saame me nelja häkatoni töötlemata vestlusandmetest. Me loome koondpaneeliga veebirakenduse esitamaks seda informatsiooni häkatoni korraldajatele.

Veendumaks selle informatsiooni vajalikkuses ja koondpaneeli kasutuskõlblikuses anti ette uuringus osalejatele erinevad situatsioonid, millega veebipõhiste häkatonide korraldajad tüüpiliselt kokku puutuvad. Osalejate interaktsioon koondpaneeliga salvestati ning sellele järgnes küsitlus. Küsitluse andmete töötlemisel selgitati välja koondpaneelil oleva informatsiooni asjakohasus ning võimalused kuidas veebirakenduse koondpaneeli paremaks muuta. Kuigi osalejad jäid üldiselt rahule veebirakenduse koondpaneeliga, oleme välja toonud sammud, kuidas koondpaneeli saaks täiustada. Uurimuse tulemusest järeldub, et veebipõhiste häkatonide korraldajatele on loodud koondpaneel kasulik saamaks üritusest paremat ülevaadet.

Võtmesõnad: Hackathon, Mõistmine, Armatuurlaud, Korraldaja
CERCS: P170 Arvutiteadus, arvanalüüs, süsteemid, kontroll

Contents

1 Introduction

Hackathons are an emerging part of not only Science Technology but also Social Landscape in the twenty-first century. We refer to hackathons as time-bounded events where participants gather to find solutions to problems or work towards producing and eventually actualizing innovative ideas [1]. A given hackathon event is oriented towards a central theme or technology such as health, economy or a more technical problem such as cyber-security [2].

Online hackathons have been around for quite some time now. However since the onslaught of the pandemic crisis, face to face activities had to limited to a bare minimum to curb the spread of virus. Restriction on gatherings during the pandemic did not stop hackathon organisers from conducting the hackathon as they resorted to online spaces for running hackathons. We can say that online hackathons are here to stay. Given the colocated hackathons were the mainstream type of hackathons, hackathon organisers had accustomed well to the ways of managing a hackathon and running it. Online hackathons, however, present numerous novel challenges to organisers that they were not exposed to before.

Co-located hackathons have certain advantages over online hackathons. Co-located hackathons afford you the ability to use your senses of hearing and sight to know what is happening around and what the other people in hackathon are engaged in. On this basis, it can be argued that it is easier for the participants to be on the same page as their team. It is also easier for organisers and mentors to know what the teams are upto and be up to date with their status. In CSCW literature, this is referred to as awareness and is especially relevant in the context of synchronous collaboration which is a work model that co-located hackathons follow naturally.

Most hackathons consists of a pre-hackathon stage (planning), the main hackathon event when teams come together and work on their ideas and post-hackathon stage [3]. Given the remote synchronous collaborative work nature of the main hackathon event, the problem of awareness persists in online hackathons. In an online setting, it is difficult for participants, organisers and mentors to know what their colleagues' engagement or availability. In CSCW literature, this problem is termed to as awareness deficiency. Although teams face the problem of awareness of team collaboration and progress, this problem is exponentially complex for organisers given the large number of teams they

have to take care of besides a host of other responsibilities (managing checkpoints, on boarding mentors and jury etc) while running hackathon event from behind a computer screen.

Hackathon organisers need to have an overview of the hackathon and in particular how the individual teams are doing. It is important for organisers to have this overview of the hackathon in order make them into more productive events. This is especially important as a lot of resources go into creating hackathon event (participants/organisers time, financial resources etc.) [4].

Most of the existing research and development on hackathons has primarily focused on co-located hacakthon events. While findings an tools from research collocated hackathons may be applicable to online hackathons, the difficulty of achieving and maintaining awareness in online hackathons is novel. Currently, there are no tools available for providing hackathon awareness to organisers. This makes it hard for organizers to retain an overview of how teams collaborate and intervene when necessary. For hackathons with a larger number of teams/participants, the number of teams/participants increases considerably making it even more infeasible to do so.
Accordingly, we formulate the following research questions:

RQ1: *What information do the hackathon organisers need in order to keep overview of a hackathon?*

RQ2: *Which information is available for keeping an overview of a hackathon?*

RQ3: *How can we present this information to aid organisers in keeping an overview of a hackathon?*

We study the relevant literature on Remote Synchronous collaboration and hackathons to identify key needs of organisers' awareness of teams. We classify these needs into three distinct use cases. In order to answer the second research question (RQ2), we explored existing infrastructure of online hackathons to find support the data we are looking. We gathered communication (chat text) data from three online hackathons to find information that can support in providing a real-team overview of hackathon. We were able to find the relevant data and turn it into information pipeline. We refer

to this as 'signals'. This is discussed in further details in 3. For answering the third research question, we develop on our findings from the second research question to propose a web-dashboard solution. The dashboard provides hackathon organisers with real time awareness information in a structured format. To complement our answer to third research question we evaluated the feasibility of data support we found for response to research question. We also evaluated the usability of the dashboard, we conducted a study where we systematically assessed use of application by professional online hackathon organisers. We particularly focused on identifying problematic aspects of the application's user interface and the general workflow of the application. Through our analysis of the study, we present further improvements that make the dashboard a totally viable product for organiser awareness. Moreover, we identity challenges that we face on further development of application and potential unexpected behaviour in unforeseen circumstances.

1.0.1 Thesis outline

In the following section 2 we highlight the existing research and developments efforts in the field of hackathons and what is the research gap particularly in relation to online hackathons. We signify the challenges of remote synchronous collaboration particularly effecting the organisers in online hackathons.

In section 3 we describe our approach on answering the research questions. We describe the steps we took based on our findings from literature. We also describe the data we collected and how we processed it to create data support for the dashboard that we developed. Furthermore, it details the study we conducted on the dashboard usability and finally we outline how we analysed the data. In essence, this section of the study outlines the literature research conducted and how we collected the data to build a first version of the dashboard. The section also describes how we conducted a usability study on the first version of the dashboard and analysed the collected data.

Section 4 describes our answer to the research questions presented earlier. In particular, we explain our findings from literature study on what information organisers want to have in order to have an overview of the online hackathon. We refer to these as use cases. We also describe how we can extract this information from the typical data sources in an online hackathon. We refer to this information as developed signals. We then describe

the dashbboard application that we came up with to present the developed signals to hackathon organisers. Finally, the results from evaluation study are discussed at length.

In the final section, we describe how the results indicate that research gap outlined in the Background section can be fixed with dashboard application. The last section sheds light on limitation of the overall study. We also briefly discuss the implications associated with the dashboard.

2 Background

Hackathons can take the form of collocated events held for a short period of time (generally over a couple of days). A colocated hackathon refers to an hackathon event where participants share the same physical space and hence collaborate face-to-face [5]. Activities in a collocated hackathon can be termed as on-site synchronous collaboration. Up until the Pandemic hit the world in 2020, collocated hackathons were the norm. Colocated hackathons are a well-studied phenomenon. There are various aspects of a hackathon that are of particular interest to researchers and organisations. It is so because hackathons are drivers innovation and possibly breeding ground for ideas [6].

Online hackathons, on the other hand, have participants functioning from geographically different locations by the means of communication technologies and collaboration technologies. Activities in an online hackathon can be termed as 'remote synchronous collaboration' according to CSCW (Computer Supported Cooperative work) literature. Given that participants are distributed across timezones and they may choose to work during the day hours or night hours. Although the participants may have to wrap up their work in the time window specified, they may not be exactly working at the same time. Hence, it may not be exactly termed as 'remote synchronous collaboration'. Still, the work style can be analogous to 'remote synchronous collaboration' as we explore later in this study.

It is important to understand the role of hackathon organiser in our study. Hackathon organisers are personnel in charge of designing a hackathon event and running it, this constitutes a vast range of tasks. These tasks can be for instance, marketing the hackathon event, organising the venue, planning timeline of the hackathon event, recruiting participants and eventually running the event [4]. The existing research on hackathon has evolved on how to organise and conduct hackathons successfully[7], often focusing on

how to tailor hackathons according to a particular goal or audience in addition to social and technical aspects of hackathon organization. Moreover, there are is a vast array of literature including guides [8, 9] and other sources such as the comprehensive planning kit [3] available for colocated hackathons.

The existing research to support hackathon organisers is extensive such as the comprehensive guides for hackathon organisers [3, 10, 8], literature such as [11, 9] and online tools such as [12, 13, 14, 15, 16]. The list of resources available for colocated hackathons may be though very comprehensive. Our study caters to the challenges posed by the online hackathons. In particular, our work focuses on bridging the gap left in the research for hackathon organiser support in online hackathons.

Online hackathons pose a challenge where organisers are geographically isolated from the participants. In a colocated hackathon, organisers are getting an overview of the state of the hackathon by capturing necessary information through visual or oral means for instance by physically walking around the hackathon venue while also perhaps asking questions to teams. In online hackathons, this is not possible because of the geographical isolation.

2.1 Awareness Research

A very comprehensive study on awareness [17] establishes the basic notion of awareness as a user's understanding of the situation including knowledge of other users and the environment that user is working, acquired actively from environment itself or through awareness technologies employed in the environment. Awareness research has evolved for over more than two decades and has been closely complemented with Technological prototypes [17].

CSCW literature provides support for awareness in Remote Synchronous collaboration. The research points on the challenges of Remote Synchronous collaboration in context of teams working Software Developments and mostly in established settings such as in tech companies [18, 17, 19]. In such a setting, the team members generally know each other beforehand for a longer period of time. Software development teams also work on topics familiar them and falling under their expertise. Hackathons, on the other hand are different in this regard. In a hackathon, team members generally do not always know each before. Teams work on unfamiliar topics often out of their expertise

and they have to basically set up their own infrastructure. This is unlike practises in a workplace where the infrastructure is already in place. Hence, the existing awareness research may not be able to cater for these differences posed by hackathon.

Remote work is an intricate process that has to be supported by a host of factors, two of the important factors being having high common ground between team members and collaboration technology [20]. High common ground refers to retaining awareness of team members. While collaboration technology refers to the tools designed for collaborative work that also aid in establishing awareness. However, such awareness research in the context of remote work mostly focuses on establishing awareness within one team. This does not cater to the problem of hackathon organisers who want to retain an awareness of how multiple teams are collaborating. Hence, the gap the in literature comes from not knowing how to support the hackathon organisers in providing an overview of online hackathons. In particular, we want to know what information needs to be provided to hackathon organisers and how can this information be gathered. This forms the motivation behind our first and second research questions (RQ1, RQ2).

2.1.1 Tools available to help Hackathon Organisers

The currently available tools [12, 15, 13, 16, 14] provide differing range of features to aid organisers in planning and running a hackathon. These tools are focused on co-located Hacakthon events. Online hackathons comprise of remote collaboration and hence there are inherent differences in the nature of an Online and co-located hackathon. Moreover, existing tools mostly help organisers in carrying out the pre-hackathon activities such as managing participant lists, listing teams and broadcast communication. In other words, these tools are not meant for running a hackathon event but rather planning.

Although, Kreativdistrikt [14] does cater to additional needs of planning and running of online hackathon, it does not cater to the problem of organizer'awareness of teams. Our study aims to develop support for solving this problem by proposing the idea of an Awareness Dashboard supported through its use in various scenarios or use-cases as we may refer to them throughout the rest of the study.

Awareness tools are a common sight in everyday life. Some common example of such technologies are Google Docs [21], Atlassian Confluence [22] and code hosting systems such as [23]. These technologies allow some way of knowing what changes were

made by whom and when. However, the usability of such tools in a hackathon setting is questionable. The reason for this twofold. First, these tools focus on a specific setting. For instance, Confluence focuses solely on document and knowledge base management and may not cater for presentation and or prototyping purposes. Secondly, these tools require people to need to use them in a certain way to be effective. GitHub works for when teams who have knowledge of git systems and have setup their remote repository, otherwise, team members cannot make us of GitHub. These tools cannot help to provide hackathon organisers with an awareness of the teams.

There is gap in the industry for tools that can deliver awareness information to the hackathon organisers. This in part may be because of gap in the literature on what information is needed for organisers of hackathon to gain an overview of the team. Hence, there has not been a need for a tool that can deliver awareness information. This absence of a tool motivates us to find means of delivering awareness information and hence establishes ground third research question (RQ3).

3 Methodology

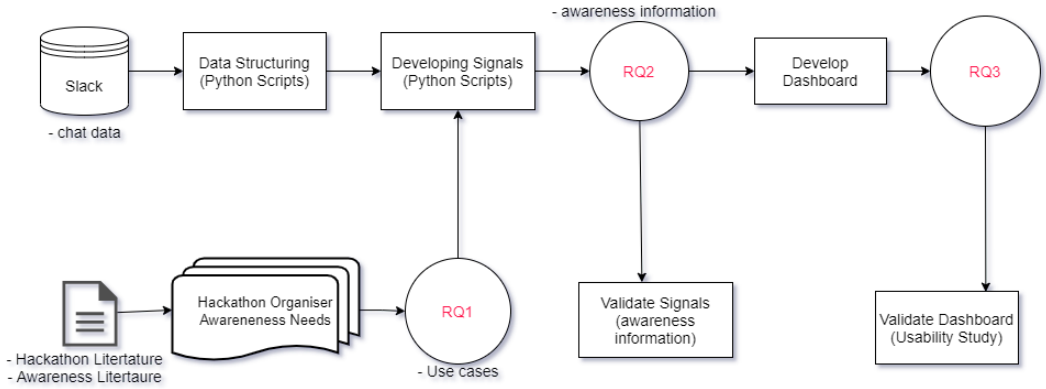


Figure 1. Study Design

3.1 Hackathon organiser awareness needs

In order to understand the hackathon organiser awareness needs, we consulted the relevant Hackathon literature and awareness literature. Awareness literature is very

broad and has progressed a lot over the last 25 years [17]. In order to answer research question one (RQ1), we had to first understand nature of hackathons and in particular online hackathons. However, there is not a lot of support available for understanding the online hackathons as they are a more recent phenomenon. To cover for this, we also studied the remote collaboration. Remote collaboration is analogous to the workings of teams in online hackathons in certain ways. Hence, with an understanding of remote collaboration, we were able to get insights into the workings of an online hackathon when studying hackathon literature. Through this we were able to identify that key characteristics of hackathons in general and online hackathon in particular for instance, teams may be needing additional members. We were also able to identify the key measures of an online hackathon that hackathon organisers care about in order to turn it into a successful event, for instance, hackathon organisers want teams to be complete in terms of team members. We consulted awareness literature given our understanding of online hackathons to understand what organiser's need to be aware of the multiple teams engaged in an online hackathon. In particular, we focused on awareness literature in context of remote collaboration. Together with our findings from study of hackathon literature and awareness literature, we were able to put together the awareness needs of hackathon organisers. Eventually we classified our findings into use cases that that helped us answer first research question (RQ1).

3.2 Developing Signals

In order to implement support for the use cases through above functional requirements, having the relevant data is a perquisite. For this purpose, we developed a comprehensive data extraction model on raw data source from actual hackathons.

In March 2020, the Estonian tech community ventured organizing a 48-hour online hackathon in an effort to develop on ideas and for creating tools that may help in combating the effects of problems caused by COVID-19. There were additional hackathons organised for addressing the same issue. The raw data used was sourced from four hackathons that were organised online between March 2020 and June 2020. In particular, there were three additional hackathons resembling each other in terms of organisation. We selected four hackathons (hackathons H1 through H4 in Table 1) for working with their data because of a number of reasons. First, all of these four hackathons used Slack

for communication. Slack was the only tool for inter-teams communication. This means that for communication from one team to another or from a team to organisers/mentors, Slack was used. Although teams were free to use any tool for intra-team communication, Slack was the recommended tool by the hackathon organisers and most of the teams ended up using it. Second, the communication channels set-up in these hackathons followed a standard structure. Each of the participating teams had their own channel for inter- team communication while there were given broadcast channels for intra-team communication. Data from the team channels was accessible as well as data from broadcast channels.

Third, all four hackathons followed a standard strategy for team formation. The teams were self-forming in all the four hackathons. This allowed the participants to form team on their own before the start of the hackathon or during the initial stages of the main event of the hackathon. Finally, these hackathons were comparable in many other aspects such as mentor support being available on on-demand basis and not permanently assigned to a team [24] but also the hackathons revolved around the same topic. Such standardized structure across the four hackathons facilitated our study.

The chat data contains information on participants, teams and messages sent by participants. An overview of this data is in figure 2 . We have considerable amount of data in terms of number and exact overview of amounts of this data is available in Table 1.

Data Pre-processing

In order for the data to be in a format that is usable by Dashboard application, we convert raw JSON chat and user information data into well-structured format. The data sourced is in raw JSON format, distributed over a large number of files specific to each team channel. Each of these team specific JSON files contains details of all the messages sent in a given team channel.

The data needs to be put into a more well-structured format in order to enable an easier access to data hence helping us in eventually answering the research questions. We use the conventional tools available to extract the data organise the data into three CSV files. The chat data from the four hackathons was extracted, cleaned and aggregated into three entities shown in figure 2.

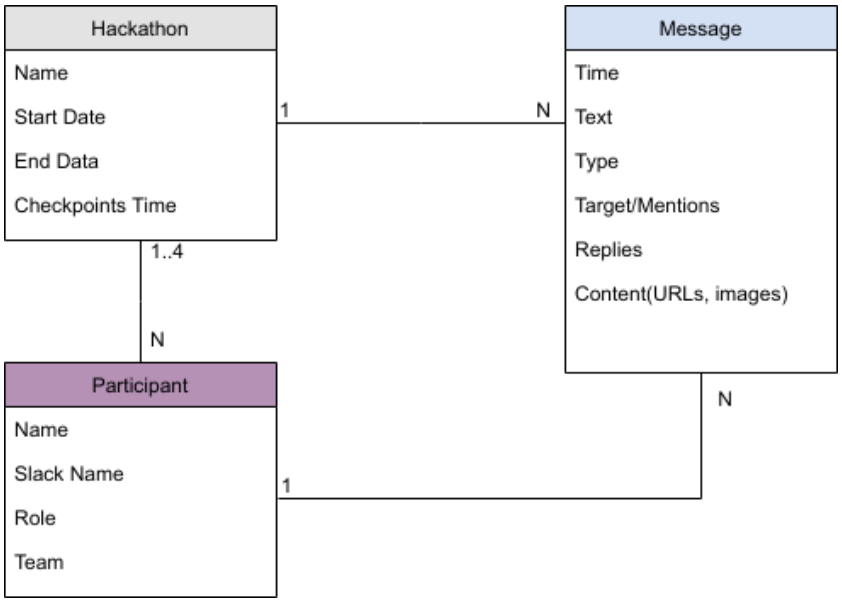
Signal means an indicator about the participants/teams that was detected through data

Hackathon name	Dates held (year 2020)	No. of Participants	No. of Extracted Messages
H1	13th March - 15th March	1508	18384
H2	3rd April - 5th April	1713	14930
H3	9th April - 12th April	4193	41485
H4	9th June - 12th June	412	4521

Table 1. Hackathons Overview

available to us by the application of certain algorithms. For instance, one of the signals was for the 'Team needs Members' functionality which referred to the fact that the back-end algorithmic structure was able to detect a team that was looking for additional members. The data was extracted for each of the three use cases describe previously.

Figure 2. Variables in the Chat Dataset



The data processing for signals development can be summarised as follows:

- Convert raw JSON chat and user information data into dataframes
- Join different datapoints to create collections for each entity (entity structure shown in figure 2)

- Identify predictive Signals and build predictive Models based on identified patterns.

Signal means an indicator about the participants/teams that was detected through data available to us by the application of certain algorithms. For instance, one of the signals was for the 'Team needs Members' functionality which referred to the fact that the back-end algorithmic structure was able to detect a team that was looking for additional members. The data was extracted for each of the three use cases describe previously.

Team Needs Members : The support for this data was developed by developing an algorithm that scans for all messages which hint towards that a team was looking for a human resource. The algorithm filters the messages which only originated from a participant and not from mentors or organisers and then looks for the team name of the originator. The indicator messages were sent either in main or especially dedicated channels of the hackathon listed by hackathon organisers at the start of each hackathon. Primarily, the messages contained the following phrases: *looking for team*|*looking for developer*|*looking for member*|*need team*|*need developer*|*need member*|*join team*|*join our team*.

Unhealthy Intra-Team Communication : The support for this data was developed by developing an algorithm that scans for all the teams that have a minimum number of members but the number of messages are less than a threshold. This threshold number is set dynamically over the period of time of hackathon's main event and varies for each hackathon. Such a low number of messages that participants are not communicating in the team's Slack channel. This may mean that a detected team may be resorting to other means of communication such as another communication platform or the team members are actually not communicating as a team.

Team Needs Help : The support for this data was developed by developing an algorithm that scans for all messages which signal that a team member was asking help in the help seeking channels. This is particularly indicated by the usage of the following phrases in text content of the message in help seeking channels: *for help*|*to help*|*can help*|*can someone*|*need help*|*external help*|*please help*|*how*|. The algorithm then then looks for the team name of the originator. These messages are sent either in main or

especially dedicated channels of the hackathon listed by hackathon organisers at the start of each hackathon.

3.3 Dashboard Development

We used the answer from the research question one (RQ1) and research question two (RQ2) for answering the third research question. In particular, we map the use cases cases (answer to RQ1) onto functional requirements given the awareness information available (answer to RQ2). The functional requirements paved the way for design of the dashboard. The dashboard development started with creating the wireframes through Figma [25] which allowed for iterative refinement of visual layout for the dashboard. The visual layout was guided by modern dashboard standards and visual components were based on a well-known UI library know as Volt React Dashboard from Themesburg [1].

The product is developed as prototype to give the hackathon organisers a complete overview of functionality that the product provides, but also to identify the unnecessary functionality earlier on and focus future development on important functionality of the product. The prototypes product allowed us to evaluate the technical feasibility of the product and enhance the functionality in a feedback focused fashion.

For realisation of **UC1**, we plan to implement as follows:

- **UC1FR1**: An alert in the landing page of the dashboard (Pop-up/sound)
- **UC1FR2**: Required action added to list of issues
- **UC1FR3**: Option to resolve/pin the issue

For realisation of **UC2**, we plan to implement as follows:

- **UC2FR1**: An alert in the landing page of the Dashboard. Teams with stating the team's absolute communication score lower than threshold will pop-up in the action required list.

[1] https://themesberg.com/product/dashboard/volt-react

- **UC2FR2**: An alert in the landing page of the Dashboard. eams with a long period of inactivity will be counted as inactive teams.

For realisation of **UC3**, we plan to implement as follows:

- **UC3FR1**: An alert for team looking for members added in the landing page of the Dashboard.
- **UC3FR2**: A page where Organiser can see a list of teamless participants with basic details.
- **UC3FR2**: Ability to resolve 1

3.4 Test Study for the Dashboard

We design a test study to evaluate the usability of the Dashboard. We particularly focused on the functionality of the application in relation to providing solutions for three use cases identified in section 3.1. Identifying problematic aspects of the application's user interface general workflow of the application was also on our agenda. This also allows us to learn about the shortcomings of the product, if any and how to improve it in future versions. The target audience are the professionals with prior hackathon organisation experience (both online and colocated hackathons). We formulate the following study goals as follows:

Study goals:

- **SG1**: Identify aspects of the workflow that may need to be improved, simplified or better explained. *this answers the question: What workflows should be focused on when designing an organiser awareness dashboard?*
- **SG2**: Identify possible convenience features that may need to be added to the application *this answers the question: What ease of use features do users expect from an organiser awareness dashboard?*
- **SG3**: Identify possible information that may be added to the dashboard. answers the question: *this tells us What additional information can help organisers in organiser awareness dashboard for a given dealing with given scenario?*

Participant ID	Total no. of hackathons (co-)organised	No. of online hackathons (co-)organised
P1	3	2
P2	11	5
P3	3	2

Table 2. Study Participant Experience Overview

- **SG4**: Identify aspects of the visual design that may need to be changed. *Answers the question: Which types of information delivery UI elements (tables, charts, alerts) should be used in organiser awareness dashboard application?*

The study was conducted in the form of online interviews with participants. The participants were selected on basis of their prior experience with organising online hackathons. Moreover, the participants were selected such that the domains of the hackathons they organised were different but also that they had different goals for their hackathons. This allowed for a diversity in the feedback received in the study. The experience of the participants is shown in table 2. The diverse experience of participants in the domain of online hackathons allowed for thorough feedback. Each of the study interviews can be divided into three distinct sequential parts each with its own purpose and expected outcomes. Those three parts are enlisted as follows:

3.4.1 Task Based Interaction Observation

In the first part of the interview, the participants are introduced to the web dashboard by allowing them to explore various parts of the dashboard. The participants are then presented with a scenario based on the three use cases identified in section 3.1. These use cases represent a scenario that hackathon organisers are likely to encounter in most online hackathons. The scenario document also enlists typical steps that organisers may take to navigate through the presented scenario.

As the participants navigate through the scenario, they are receiving information about the state of a hackathon and consequently perform related actions on the application. Doing so required understanding what particular information indicates, understanding

what certain UI elements are for, understanding how to execute the provided tasks and correcting mistakes. The participant's entire interaction with the App is recorded until the steps are all completed. This interaction contains part of the information that we are looking for as part of the study.

In order to understand the participants interaction experience with the dashboard, we employed Nielsen's usability heuristics that serve as a good basis for what to look out for during the participants interaction with the application [26]. We were able to particularly focus on potential issues that the participant had using the application. According to underlined heuristics the included the following:

- Visibility of system status (Delivery and Readiness of Awareness information.)
- Match between system and the real world (Match between dashboard and the real hackathon dynamics.)
- User control and freedom (User interaction and freedom)
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

3.4.2 Interview Questions

The first part of the interview was followed up with detailed questions focused on usability of dashboard in the scenario based on the three use cases from section 3.1.

The questions can be classified into three distinct categories as follows:

- UI Related - such as :

  - Which UI elements are unfitting or misleading?
  - Which types of information delivery UI elements (tables, charts, alerts) should be used in various places in the dashboard?

- Workflow Related - such as
  - Which workflows should be focused on more?
  - How can we possibly improve the existing workflows?
  - How can the data support/signals for the required information in the dashboard be generated?
  - What are potential problems with approach we used for generating the data signals?

- Additional Features / Information Required - such as
  - What additional information did they expect from the system for a given use case?
  - How would they like to see additional information?
  - Which ease of use features do users expect from an organiser awareness dashboard

- General - such as
  - Ideas for product expansion?
  - Which part of the dashboard is not well-built needs immediate improvement?
  - Which part of the dashboard is well-built and may not need immediate improvement?
  - General comments/suggestions on the dashboard

3.4.3 Study Analysis

Finally, We used affinity diagrams to analyse the data collected from interaction observation and answers to questions from study explained in the previous section. Affinity

diagram are used for grouping common experience through simplified drawings and are a common tool in research where the objective is to analyse qualitative data.

For this study, affinity diagramming was a suitable approach as it allowed us to gather insights about the experience of the participants from the study. Affinity diagramming let us synthesize information without subjecting us to predefined notions. This helped us get an overview of the ideas,wishes, perceptions and problems of the participants regarding our app. This is important for us as these participants (Hackathon organisers) are the prospective users of our application. The figure 3 shows a screen capture from our affinity diagramming. We used Miro Online Whiteboard [27] for the purpose of affinity diagrams.

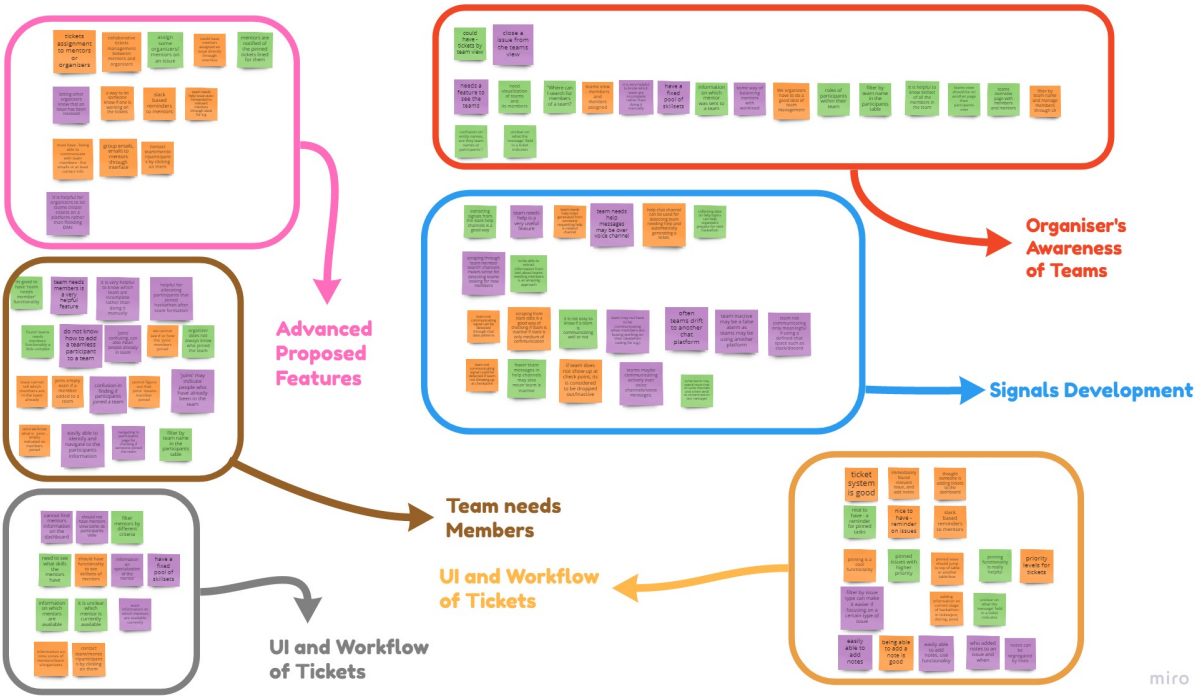


Figure 3. Affinity Diagramming for Evaluation Study

4 Findings

4.1 Hackathon organiser awareness needs

Teams Needs Help

One of the key characteristics of synchronous interactions in remote teams is the possibility of rapid feedback. Rapid feedback allows for quick corrections and removes misunderstandings and disagreements [20]. Work related technical discussions is the primary theme of conversation in group chats at workplaces. Hence, it is highly likely that technical questions may arise here and participants look for help in the team chat. Therefore, team channels are the place to check for teams looking for help and asking questions [28]. Teams with mentor support do particularly well and this support is particularly appreciated when it is requested for by the participants than provided when there is no need. One way this can be achieved is through Organisers checking for team channels for any indication of help/mentor input needed and then directing the mentors to the required channel [24]. Hence, teams should be provided with mentor help when there is a perceived need by the team participants. In an online hackathon setting where text chat is the primary medium of communication, it is logical to say that organisers should be aware of questions asked in the teams channels so that organisers are able to provide them with help and or feedback. Accordingly we propose a use case requirement:

UC1: *As an organiser, I want to know which team is looking for feedback or help so that I can direct a mentor towards the team.*

Unhealthy Intra-Team Communication

Nuanced information is a key characteristic of synchronous communication in distance teams. Continuous and analog flow of information carries itself with many subtle dimensions leading to very small differences in meaning to be conveyed and information flow is modulated [20]. We also know that if people think that their teammate is in a distant location, they initially cooperate less than if they know the teammate to be close, yet, cooperation increased with interaction [29].

We found that distributed developers do need to maintain awareness of one another, and that they maintain both a general awareness of the entire team and more detailed

knowledge of people that they plan to work with. This awareness is maintained primarily through text-based communication (mailing lists and chat systems) [18]. Preliminary research of the interface indicates a positive association between the visible number of individuals that share an activity interest and one's willingness to initiate organization of the activity [19].

Slack Communication is not easy for young people [30]. Hence, teams should be communicating in a continuous and analog fashion with as many team-members involved. Awareness of active team members promotes real-time interaction among the team members. This in turn leads to higher workspace awareness essentially meaning more well-informed teams [17]. A hackathon organiser would like to maximise the teams that are active and hence minimise the teams with inactivity/inactive members so that hackathon can be productive to the maximum. Accordingly we propose a use case requirement:

UC2: *As an organiser, I want to know which teams are not communicating (i.e. Have been inactive/communicating hardly) so that I can take appropriate action.*

Team Needs Members

Organisers have to be aware that some teams may not be completely formed at the beginning of the hackathon or some participants may show up without a team [11]. Lack of team members with required skill can heavily impact a team's performance and the product prototype that team ends up with. Hence, Organisers are motivated to provide teams with members that have the required skill set [5]. Participants often come up with project ideas and list out the roles they require in a team to work on the proposed idea[6]. Organisers often task themselves with keeping track of all the required roles/skills in various teams and hence act as a central point of contact when teams are looking for new members [30]. Hence, organisers have to facilitate teams/participants in team formation or completion. Accordingly we propose a use case requirement:

UC3: *As an Organiser, I want to know which teams are looking for an additional member, so that I can help them find a member.*

4.2 Developed Signals

In this section we outline our findings for answer to research question two (RQ2). RQ2 referred to on developing signals from existing data for helping hackathon organisers to keep an overview of the hackathon. In particular, we wanted to develop information for providing information support to the use cases found as answer to research question one in section 4.1. Our findings in this section are founded on the methodology crafted in section 3.2.

We were able to extract the desired signals for supporting use cases from the Slack chat data. A quantitative overview of the developed signals is give in the table 3. The table shows the number of signals developed for each use case per hackathon (hackathon H1-H4). Generally, the number of the signals for each use case appears to be somewhat dependant on the size of the hackathon. The bigger the size of the hackathon, the greater the number of signals developed with the exception of H3. This difference can be attributed to the difference in some structure differences in H3. H3 was split into multiple tracks where each tract referred to a particular theme that all the teams withing that hackathon concentrated on. Hackathons H1,H2 and H4 followed a somewhat similar structure and were not split into tracks.

	Signal Type		
Hackathon Id	**Team Needs Members**	**Team Needs Help**	**Unhealthy Team Communication**
H1	9	51	13
H2	96	88	107
H3	18	33	211
H4	34	95	58

Table 3. Developed signal count per hackathon

4.3 Dashboard Development

We followed the methodology outlined in section 3.3 to develop the dashboard. Figure 4 shows screenshots from the developed dashboard. The source code for the dashboard is hosted in a publicly available code repository [2]. A deployed version of the website is

[2]https://github.com/abdul0214/HackDash-UI

available. [3]. However, the dashboard is currently an only frontend application with some sample data filled in. The backend application is expected to provide data persistence and integration with additional platforms such as Slack for automatically developing signals during a hackathon event according to the methodology described in 3.2. The current version of the dashboard application allows for user interactions in a fashion similar as it would with a backend. We used this current deployed version of the frontend application for our evaluation study. + Figure 4 shows screenshots of three major components of the dashboard. Figure 4a shows an issues table where all the issues detected with the backend system are displayed in tabular format. The tabular view also shows some basic information about the ticket such as the ticket type, name of the team that issue was detected with, time of the issue detection and current status of the issue along with issue ID. The table also currently allows to filter the issues by their status through the select box on the top right side of the table.

Upon clicking one of the issues in the table, the modal view in figure 4a pops up and displays further details on the issue along with an options to perform given actions on the issue. Among the detailed information available is the 'message' field that refers to text message from the team based on which the issue was detected by the backend system. There are additional fields such as the 'joins' which refers to any new participants that have joined the team after the issue was detected with the team. This view also allows the hackathon organisers to add any notes to the particular for referring to it later on. The notes are then displayed in the same view. The 'RESOLVE' button allows the hackathon organiser to change the status of the issue to resolved the issue while the 'PIN' button allows the hackathon organiser to change the status of the issue to 'pinned'. The organiser can also choose to not take any action and simply close the issue by clicking on anywhere except the issue or clicking on the close icon on the issue.

Figure 4c is a tabular view of the information on participants and mentors. The table entries display basic information on each of the participant. Unlike the issues table in figure 4a, the entries of this table are not clickable. The 'Filter table' checkbox allows us to filter the participants by various criteria. This functionality allows us to view only mentors or participants that are team-less. This table is expected to help organisers in finding team-less participants so that they can help out a team looking for new members. The table also allows organisers to filter only mentors direct them to the teams that maybe

[3]https://hackdash-ui.web.app

looking for help.

4.4 Study

Feature Validation/Confirmation:
Explains the perceived usability of the feature by the users(study participants) given the typical hackathon scenario presented earlier.

4.4.1 Development of Signals:

Validation:

In this section, we discuss findings from the evaluation study that are related to research question two (RQ2).

Team Needs Help

As discussed earlier in section 3.2, we developed the team needs help signals by building crawlers for slack chat data to detect signals of team looking for help. Participants to the study agreed that extracting signals from slack help chat data channels is a good to extracting information for slack help. *"extracting signals from the slack help channels is a good way." - P3*, *"Help chat channel can be used for detecting team needing help..." - P2*.

Team Needs Member

Signal for a team that is looking for members was pertinent to the development of 'Team needs members' awareness functionality. Participants agreed on choice of approach for the development of this type signals that these signals can be developed based on finding relevant chat information in channels that are for finding new members. The exact approach for development of this type of signal is explained earlier in section 3.2. Participants were found to be saying: *"scraping through 'team member search' channels makes sense for detecting teams looking for new members" - P1* while another participant found the approach as creative *"to be able to extract information from text about teams needing members is an amazing approach" - P3*.

Unhealthy Intra-team Communication

For this functionality, we developed a system to look through team channels to find out which teams may not be communicating well. More details on how this signal

was constructed is explained in the earlier section of methodology3.2. A participant acknowledged the approach to this developing this signal from chat data as follows *"team not communicating signal can be detected through chat data patterns" - P2*. At another point a participant also referred to this as the right approach if slack messages are the only medium of communication *"scraping from slack data is a good way of checking if team is inactive if slack is only medium of communication" - P2*.

Problems associated with Signal Detection:

Although participants did not mention any potential implications of using slack chat data for signal detection for 'Team needs members' and 'Team needs help', participants did highlight potential issues with using slack chat for detecting 'Unhealthy inter-team Communication'. Participants mentioned a host of potential problems related to detecting unhealthy team communication.

One of the most common concern among the users was that not all teams are using Slack for inter-team communication. It is not uncommon for teams to use a platform other than Slack for text communication. In such a case, the feature will be producing a false alarm. *"It is not easy to know if a team is communicating well or not" - P3 "Team inactive may be a false alarm as teams may be using another platform" - P1. "Often teams drift to another chat platform" - P1*. Let us refer to this as **Issue S1**.

Another problem pointed out by the hackathon organisers was the growing tendency to use voice messages for inter-team communication. *"teams maybe communicating actively over voice channels/voice messages." - P1. . "some teams may spend more time on voice channels and others tend to concentrate on text messages" - P3*. A system relying solely on text messages may not be able to assess the team communication pattern correctly. Let us refer to this issue as **Issue S2**.

Moreover, one of the users also highlighted that at times the team communication maybe little to non-existent especially when team members may be involved in working on their assigned tasks related. *"Team may not have to be communicating when members are busing working on their idea(when coding for e.g.) - P1* However, this is less likely to make the backend system detect the concerned team as having unhealthy communication.

Response to Problems:

We can propose a logical fix to the problem posed by **Issue S1**. Hackathon organisers can define the chat communication medium for inter-team communication. This medium can be one of the popular applications such as Slack or Discord. This idea was also

brought forth by one of the users as a comment *"team not communicating only meaningful if using a defined chat space such as slack/discord" - P1*. Using a defined workspace will allow the backend system to avoid false positives as it will have the chat data to detect unhealthy team communication.

Issue S2 poses a technical implication of solely relying on text messages for assessment of inter-team communication. In order to avoid this issue, we can modify the backend system to include voice messages in the assessment of inter-team communication. Although voice messages are also used in public channels(non-team channels) they are very infrequent. Chat in public channels such as channels for asking help is mostly restricted to text messages. Our backend system will still be able to detect the signals for 'Team needs help' and 'Team needs Members' since backend system scans the public channels for these signals.

Further Ideas:

During the study interview, additional ideas were brought forth by participants. One of group of ideas was related to additional ways of detecting unhealthy team communication or an inactive team. Fewer or no messages by a team in public channels such as general channel or channels for asking help may indicate that the given team is inactive. *"Fewer team messages in help channels may also mean team is inactive" - P3*.

It was also brought to attention that analysing communication by teams in general or checkpoint channels around the time of checkpoints can provide insights into the whether the team is active or not. *"If team does not show up at check point, its is considered to be dropped out/inactive" - P2* . The functionality for 'Unhealthy Team Communication' can be supplemented with the addition of team communication detection around the time of checkpoints.

4.4.2 Team Needs Members:

Feature Validation:

The analysis results indicate that Hackathon organisers expressed a desire to have such convenience feature and found such a feature to be useful. While not all organisers agreed on the usefulness of such feature. Two participants quoted as follows:

"its good to have 'team needs member' functionality" - P3, *"team needs members is*

a very helpful feature" - P1 and *"it is very helpful to know which team are incomplete rather than doing it manually" - P1*.

One may argue that some hackathons have self-forming teams and hence usability of such feature is limited in such a situation. We found out that even in cases where teams are formed in preparation stage of hacakthon, there may be still incomplete teams and more importantly there can be be participants joining hackathon after this stage. Hence, this functionality can be helpful in situations to allocate new participants to teams where teams are looking for new members. *"helpful for allocating participants that joined hackathon after team formation"- P1*.

The workflow for this feature relied on two parts of the dashboard: the issues/tickets table on the landing page and the participants table on the participants page of the dashboard. It was observed that participants were able to identify the purpose of both components without any information or guideline required from the interviewer.

Problematic Aspects of the Feature:

However, the functionality can be little complex to understand at first. *found 'teams needs members' functionality a little complex - P3*. One of the most problematic aspects of the this functionality was with how the required information was delivered to the organisers. Participants found the 'joins' information (a field in the issue of type 'Team needs Members') to be confusing in different ways. A participant reported this functionality as difficult to understand because of the term 'joins' , *" 'joins' can also mean people already in team" - P1. " cannot figure out that 'joins' means member joined" - P2*. Let us refer to this as as **Issue N1**.

Participants expressed that organisers may not know on who joined a team after the issue was detected. *"organizer does not always know who joined the team" - P3 "confusion in finding if participants joined a team" - P1. "joins empty even if a member added to a team" - P2*. The issues pointed out by the participants are valid as our backend system may not be always able to detect if a new participant joined the team that was needed by the team.

Relating to the above issue one may also argue that if another organiser in the same hackathon had already asked a participant to join the team, the organiser checking the issue now will not be able to know and there may be multiple organisers engaged on the same issue. *"We cannot see if or how the 'joins' members joined" - P2*. Such a duplication of organiser efforts in undesirable as resources(organisers time) are of high

value. Let us refer to this as as **Issue N2**

Moreover, the participants pointed out the lack of information on the current structure/formation of a team that was looking for additional members. *"... cannot tell which members are in the team already" - P2*. Users expressed that they want to see the members that are already in the team. *"issue cannot tell which members are in the team already" - P2*. Hackathon organisers want to be aware of the members that are already in the team. This information was not provided on the dashboard. Let us refer to this as **Issue N3**.

Besides, users perceived the dashboard to provide support to add participants to the team. *"do not know how to add a participant to a team" - P1*. This functionality is not offered by the dashboard. The dashboard is supposed to provide awareness to organisers about which teams may be looking for members and any other relevant information required. Nonetheless, we would like to address this issue further and we can refer to this issue as **Issue N4**.

One of the issues observed during interaction of the users with this functionality was that users were not sure if the participants they are choosing for adding to a given a team has the right skillset that the target team is looking for. Hence the user suggested to have skillset information of the participant in the participant view. *"I want to be able to see the skillset of the participant" - P2*. This is a valid concern for organisers as they want to add the members to a given team based on the needs of the team. If a team is lookinf for a new member with graphic designing skills and the organisers adds someone with software engineering skills, it is of little help to the team. Let us refer to this issue as **Issue N5** and propose a solution to it in the upcoming section.

Response to Issues:

Issue N2

The second issue is about organiser not being aware if a participant joined the team that was looking for help or which participant joined the team if any. This issue is more complex in nature. The root cause of this issue is two fold. In the situation when a participant is added to the team either by organiser's action or through any other means, the 'joins' field may still be empty, indicating that no new participant has joined the team. This can be confusing for many users. In an another situation when a participant joins a team without any direction from organiser, the system may not be able to detect that a participant has joined the team.

Once participants understood what the 'joins' field means, they were able to understand that empty field means no new participants have joined the team. *"Once we know what is 'joins' , empty indicated no members joined" - P2*. Hence, part of the solution to the problem where users are not able to understand if new participants have joined the team or not is manifested in the solution to the **Issue N1**.

Without any fixes, we can propose the users to use the existing functionality of adding notes to the issue. As said earlier in a situation where a participant joined the team on organisers directions or request, the organiser can add notes stating something simple about a participant joining a team. This note can be something as simple as 'Participant X joined the team' where X is the name of the participant. Alternatively, we can propose an added functionality to this issue in the next section. The fix to the Issue2 for the second situation may be difficult to implement as yet and requires for more technical research. However, once an organiser finds out that a participant has joined the team, organiser can add a note about the participant joining the team just like as explained earlier.

Issue N3

This issue concerns with user being able to see the team structure especially when viewing the 'Team needs members' issue. It is clear that this functionality is much desired from the perspective of hackathon organisers. Currently, the dashboard does not display such functionality. Support for such a functionality was included in the dashboard. An simple idea for viewing the participants of the team was observed from the comment of one of the participants during the interview. *"I should be able filter by team name in the participants table" - P3*. According to this comment, users should be able to filter the participants from the table based on their team name. This filter will provide us with all the participants that are in the team name filtered for. Hence, the result of this filter can provide support for Issue N3, letting users have an overview of the team.

Issue N4

Issue N4 relates to a desire of the users to be able to add the participants to the team directly through the dashboard. The exact implementation of this issue is a technical task requiring development effort to create an infrastructure integrated with chat platform and/or any other team management platform. This issue does not fall exactly under the context of providing awareness to hackathon organisers. We may discuss more about this issue in the upcoming sections where related topic may be discussed. In this section

we do not have concrete response to this issue.

Issue N5

Although thi issue of great significant to the 'Team needs member' functionality, there may not be a elegant of way of supporting this given our dashboard design. Having an auto-auto-detectable skillset functionality in the backend system is perhaps a fat-fetched functionality. However, a workaround may be to use Lastname field of the user's indicating their skillset chosen from a fixed pool of skillsets. The backend system can scan this and display the skillset of the participant in the participants view.

Improvements to be made:

Issue N1

It is clear from the comments of the participants that they were not able to understand the meaning of 'joins' field in the Team needs Members issue interface. Hence, we can resolve this issue by changing the name of the field to a more meaningful name such as 'New members'. Additionally, in order to make it even more intuitive we can add a tool-tip giving a brief explanation of what this field means so that when a user(hackathon organiser) hovers over this field. The tool-tip text will become visible, making it easier for the user to understand what the field is about. The tool-tip text can be as follows: 'This field indicates the members that joined after the team was detected to be looking for additional members'.

Issue N2

For additional support to Issue N2, we can propose a solution that can be more intuitive rather than adding notes for when a participant is added to the team. We propose a solution in which the 'joins' or the newly proposed name 'New members' field can be made an editable field. Hence, when an organiser adds participant to the team manually, the organiser can edit the field. This will the organisers to be conveniently aware that some participant has been added to the team.

Issue N3

The issue of being able to see team members is commonly reflected problem throughout the study by almost all the participants. One solution to the issue was proposed by the study user them-self when they mentioned that team name filter on participants table will allow them to see the members of a team. We can consider to implement this solution. However, at other places throughout the interviews, users expressed a desire to have a

separate component on the dashboard for viewing the structure (participants + assigned mentor if any) of the team.

4.4.3 Team Needs Help:

Team needs help feature overlaps with organiser's awareness of mentors refers to organisers awareness of the mentor resources. In particular, organisers want to know the basic details of mentors, their skillset, their availability and their current engagement with any team.

Problems:

Users pointed out on having not enough information on dashboard regarding mentors. *"cannot find mentors information on the dashboard" - P1*. Dashboard provided a list of mentors with some basic information. This included very basic information : name, time since join and repeat participation. Information on mentors was provided on the same view as Hackathon participants. However, users did not like this approach and suggested having a separate view for information on mentors , perhaps on another page of the web dashboard. *"Should not have mentors view same as participants view" - P1* . Lets us refer to this issue as **Issue M1**.

All users consistently pointed out need of having information of specialisation or skill-set of mentors. *"information on specialization of the mentor" - P1*. *"should have functionality to see skillsets of mentors" - P2."need to see what skills the mentors have" - P3*. Organisers' want to have this information for resolving help requests from teams. Teams maybe needing expert help on some given topic. Organisers want to send those mentors to the team who have skill-set in the given topic so that they are able to help the team. Lets us refer to this issue as **Issue M2**. User's suggested that we use a fixed pool of skill-set. *"have a fixed pool of skillsets" - P1*. This can allow for consistency and hence easier understanding of the skillset of the mentor by the organisers.

Besides information on skillset of mentors, organisers felt lack of information on availability of mentors at the given instance. *"it is unclear which mentor is currently available" - P2*. Hackathon organisers want to have information on the availability of mentors.*"Need information on which mentors are available" - P3*. *"want to know which*

mentors are available currently" - P1. Let us refer to this issue as **Issue M3**

In online hackathons, participants and mentors are distributed over large geographical distances and so it is common to have time zone differences. Organiser's want to assign a mentor to a team with lesser time zone difference between timezone of team and mentor. *"information on time zones of mentors/teams/organizers" - P2*. Let us refer to this as issue **Issue M4**

Users highlighted that in cases of large online hackathons, the number of mentors is large and hence there should some way of filtering mentors on some criteria. The filtering can be done a range of different information such as time-zone,skill-set of mentors.*"need to filter mentors by different criteria" - P3*. This can work definitely since we have a given set of timezones to choose from and also a fixed pool of skillset as explained earlier . Let us refer to this as issue **Issue M5**.

Solution to Problems:

It is understandable for users(hackathon organisers) to have information of mentor resources. This is especially since the purpose of the dashboard revolved around providing awareness to hackathon organisers. However, organisers find Mentor information to be of a different significance and hence request the mentor information to be presented in a view separate from participants. To cater this issue(**Issue M1**), the next iteration of the website mentors information can be presented in a separate tab on the dashboard with link in the navbar of the dashboard.

Mentors primary role in a hackathon is to provide expert help to participants in the hackathon. Teams may be needing expert advice or help in one of the various areas such as Graphic Designing, DevOps, Business Management etc. Organisers are directing mentors to different team. Organisers need to match the skillset of the mentors with the area of support the team may be needing. This essentially means that organisers need to be aware on the skillset of the mentors before they can direct a mentor towards a team. However, our backend system cannot detect the skillset of the mentors. However, we can alternatively propose a system where organisers are allowed to enter information on skillset of mentors as dropdown field of the mentors at any stage during the hackathon.

Once we have information on skillset of mentored into the system, it can be visible to all the organisers throughout the hackathon. Although we do not provide a sophisticated solution to the **Issue M2**, the solution can offer support for the awareness of mentor's skillset to organisers.

Issue M3 refers to organisers not able to find out if a mentor is available currently. A mentor may not be able to attend to attend to team's looking for help for various reasons such as personal reasons or mentor being busy in mentoring another team. Based on their experience, the hackathon organisers have stated that it happens often that mentors maybe distributed over various timezones and hence may not be available always. It is also common to see that mentors are involved in some thing else also alongside their commitment as a mentor. While a mentor may also not be available because the mentor is already engaged in helping out a team. Due to these factors, mentors may not be available or may not be willing to help team at a given time. Hence, organisers want to know if the mentors are available and willing to contribute at a given time. Hence, we propose a s simple solution for information on whether or not mentors are available to help. This can be done by providing an 'availability' next to information of each mentor. The source of this data will be Slack/Discord where the mentor can set his status as 'Available' when mentor is willing to help and is not engaged with another team. Otherwise, the mentor can set the status to something like 'Away' indicating to organiser that the mentor may be engaged with another team or is not able to contribute currently.

A closely related to availability(**Issue M3**) is the issue(**Issue M4**) of timezone information of mentors, organisers want to know the timezone of the mentors. This can allow organisers to find to match the team and mentors according to timezone in addition to skillset. Having information of timezone is important, as a mentor may not be available to help a team during the odd hours of the day for e.g. night time. Similarly to solution for availability **Issue M3**, we propose a solution where we add an additional field to mentor information for the timezone. The timezone information for a mentor is fetched from Slack. Slack provides option to select/update timezone preferences to users and hence this information can be fetched by our backend system and presented on our dashboard.

Issue M5 is more related to convenience feature in visual presentation of the mentors

information. Users want to be able to filter mentors on various criteria such as the information on availability,timezone, skillset. This can be achieved through the use of filters on the view of a mentors or through use advanced search features. Given conventions of modern UI standards, this can be fairly achieved through putting the option to choose filter on top of each column in table view of mentors. Hence, this also guides us in the UI design for presentation of mentor information. A table view of the mentors can satisfy solutions to all of the issues discussed above.

4.4.4 Organiser's Awareness of Teams Structure:

Organiser's awareness of mentors refers to organisers awareness of the team formation. Organisers want to know the basic details of every team. These details include things like team name, number of participants, name of the participants. In the ideal scenario organisers would like to know further details such as roles of team members in a given team and their skills. Such information falls under the umbrella of awareness of organisers about the teams. Such information is expected to allow the organisers make more informed decision on how to resolve issues with teams.

Problems:

During the interaction with the dashboard, it was observed that organisers were not always sure of what the entity names were referring to. At times, users mistook team names for participant name and vice versa. This concern was also voiced by one of the organisers. *"I am confused on names, are they team names or participants'?" - P3*. It is important for the users to be sure of what the names of various entities refer to. us refer to this concern as **Issue C1** and get back to it later in the upcoming section.

The concern brought forward most frequently was related to having information on team formation visualisation. At many times during the interview study, users wanted to see teams and navigated to different parts of the dashboard to find members in given team. After not being able to find participants of a given team in participants table, one of the users asked *"Where can I search for members of a team?" - P3*. Upon not finding the relevant information, users did not hesitate to voice their concern on their need to see the structure of teams. *"I need a feature to see the teams" - P1. "We need visualization of teams and its members" - P3. "Need teams view - members and mentors assigned" -*

P2. Organisers stated that need for this feature also arises because during the course of a hackathon, organisers have to do a great deal of team management (assigning mentors, assigning new members etc.). *"We organizers have to do a good deal of team management. " - P2*. Let us refer to this as **Issue C2**. The organisers would like to see the skillset of the participants in the team. *"It is helpful to know skillset of all the members in the team" - P3*.

In addition to having a separate view for team, users would also like to have issues related to a team visible in the teams overview information. *"could have tickets by team view" - P3*. This would allow the hackathon organisers to have a holistic overview of the team and any problems that the team might be facing. Furthermore, users requested to have a functionality to not only view issues of a given team in the overview of that team but to be also able to take relevant actions(such as 'resolve') the issue from the same place.*"close a issue from the teams view" - P2*. Let us refer to this as issue **C3**.

Solution to Problems:

From **Issue C1** description above, it is clear that labelling of entity names was not intuitive enough for the users to understand what they mean easily. Naturally, a better naming strategy will help allevaiate this problem. We propose a entity specific names for the fields. For instance, the fields that represent the name of team should have a label of 'Team name' and fields that refer to participant or mentor names should have names such as 'person name'. In addition, we can introduce a little tooltip that will provide small explanation of what the field means. In our case, upon hovering over the relevant entity field name, the tooltip will display that the field refers to the name of an entity, for e.g. 'Name of the team in Slack'.

Issue C2 requires us to provide UI support for viewing the team. There a couple of things the users highlighted that they would like to see on the dashboard. One of the most basic things whey would like to see is an overview of team with basic information such as team name, number of participants and names of participants with any mentors currently assigned or engaged with the team. *"Teams overview page with members and mentors" - P3*. In particular, users want this team view on a page/sections separate from the participants view. *"teams view should be on another page than participants view" - P3*. As a part of the functionality, organisers also want to search the team by name and see its overview. *"filter by team name...." -P2*. One of the other user also wished for a

functionality with filtering by team name but in the participants table. However, it is is clear that participants want to filter the team from the list by team name. *"filter by team name in the participants table" - P3*. The need to view the skillset of every member of team can be satisfied in the same way as displaying participants skillset as referred to in **Issue N5**. The backend system can scan this and display the skillset of the team member right next to member details in the teams view .

Issue C3 reflects a want of the organisers to see all the issues (such as 'Team needs help') of a particular team together with its other information (team name, members info. etc). In order to provide support for this issue, we do not need additional data. This problem is more of a UI design problem and hence it can be catered through alterations in our UI design. We propose to show all the issues of a team under a team's together with team overview. According by modern design standards, these issues can be presented on cards based UI elements. In addition, these cards can contain additional information regarding the issue that is standard part of an issue. More essential to this functionality is perhaps having the ability to take the given set of actions on the issue. This will allow the users to resolve the issue and or pin it or perhaps add notes to it just like doing it in the original issues view.

4.4.5 UI and Workflow of Tickets:

This feature refers to the Ticket/Issues component on the landing page where all the issues detected with teams are displayed in a tabular view. This component maybe the focal component of our dashboard as it may the starting point for an organiser's awareness of the state of the hackathon.

Feature Validation:

Users found the feature to be intuitive and easy to use. *"ticket system is good" - P2*. *"I immediately found the relevant issue" - P2*. However,users had trouble understand the source of these issues as they considered the issues to be created by human being (participants/other organisers). One of the functionalities offered by this feature was to to be able to add notes upon clicking open the issue. User pointed out to the usefulness of the feature as *"being able to add a note is good" - P2*. Users found it easy to add notes to the issue. *"It is very easy to add notes." - P1*.

The table view also offered an functionality to pin an issue. In its currently state,

pinning the issue changes the status color of the issue from yellow to red in an attempt to highlight a greater attention needed for the issue. Users did find the pinning feature to be intuitive. *"pinning is a cool functionality" - P2*. *"pinning functionality is really helpful" - P3*.

Problems:

The users want to know who added notes to the issue and when.*"I want to find out who added notes to an issue and when." - P1*. This helps organisers get an awareness of the state of issue by knowing who is dealing with the issue and looking at the notes in a chronological manner. For further details on the designs, one of the users already provided a hint to put separate notes on separate line and not as current implementation of comma separated sentences. *"notes can be segregated by lines" - P1*. Let us refer to this as issue **T1**.

Organisers commented on the pinning functionality further and pointed out that pinned notes should be able to have a sense of elevated priority by either moving them to a separate place on the page or pushing them to the top of issues list. *"pinned issues with higher priority" - P3*. *"pinned issue should jump to top of table or another box" - P2*. But generally, it was also brought to attention to have a priority level strategy for tickets with different status. Let us refer to this as **Issue T2**.

Users mentioned that often they are so preoccupied with their duties as hackathon organisers that they tend to forget about a particular issue that a team may be facing. It is desirable to have functionality that can set a reminder about an ongoing issue to hackathon organisers. Organisers want to be able to set a reminder for a given issue.*"It will be good to have -reminder on issues" - P2* . *"nice to have - a reminder for pinned tasks" - P3*. It is understandable to have reminders especially when you want to get back later to unfinished issues that may be needing another check by organisers. Hence, we want to consider this later, let us refer to this as **Issue T3**.

Among other functionalities that organisers thought could be good for the dashboard are the having the functionality to filter the issues. *"filter by issue type can make it easier if focusing on a certain type of issue" - P1*. Let us refer to this issue as **Issue T4**.

The user's did not understand what the 'message' field in a ticket refers to. *"unclear on what the 'message' field in a ticket indicates" - P3*. This field refers to the message by team that is relevant to the issue. For instance, in a ticket of type 'Team needs help', 'message' field displays the text message which was sent by team in which they requested

help. This message can be look something like 'Can someone please help us finding website deployment options?'. Let us refer to this issue as **Issue T5**.

User want to have the information on the current stage of the hackathon. *"... information on current stage of hackathon in tickes(pre, during, post" - P2*. This essentially refers to if a dashboard is to support multiple hackathons simultaneously, then perhaps it is good to have information hackathon name and hackathon stage information in each issue/ticket. The hackathon stage mainly refers to current state of the hackathon, it is considered to be pre-hackathon if the main event of the hackathon (where all come together to bring up ideas and prototypes) has not taken place yet. During hackathon stage refers to the man event i.e. when people come together (either physically or online) to brainstorm ideas and work om prototypes. While post-hackathon refers to when the main event has already taken place. Let us refer to this issue as **Issue T6**.

Solution to Problems:

Issue T1. This solution requires UI design changes. No more additional data or backend infrastructure is required. In order to accommodate the user's wishes, we can propose a redesign where we show which organisers put in the note together with time. The new solution can be broken down into three parts. First, the system will add the name of the organisers who added a note. This name will be added at the beginning of the note. Second, there will also be a time information added towards the end of the note. This time information will be timezone adjusted. Lastly, the notes will be split on separate lines, sorted in the order of descending time. This means the top line will contain the latest note and last line will contain the earliest note if any.

Issue T2 In its current implementation, pinned notes do not have any priority in the dashboard except their status color being changed to red once they are pinned. Taking user's wishes into account we propose a addition to tickets table. This new functionality is supposed to give priority assignments to tickets/issues in table. It is important to note that this new functionality will add priority management to issues with all types of status and does not only apply to issues with pinned status. This new functionality will have highest priority for pinned issues, while open issues will have medium priority and resolved issues will be pushed to the end of the table. Within a group of issues if similar status, the sorting will be based in the order of ascending time. Among two pinned status, the issue detected earlier will be displayed higher above in the table.

Issue T3 Implementing reminders into an issue can done in various ways. The solu-

tion to this problem can have varying levels of complexity depending on the requirements of the users. Implementing a system where organisers are notified on their mobile devices may require additional infrastructure and higher development efforts. However, a simpler solution will be to add a functionality where organisers can set countdown reminders on an issue. Upon countdown getting over, the users are displayed with a pop with the issue details that they had set the reminder for. This solution satisfies the users wishes and does not require neither additional infrastructure not additional data.

Issue T4 Being able to filter various data is a repeated concern for the users during our study. In this particular issue, users want to filter the issues in the issues table based o the issue type. This is convenience feature and makes total sense especially in a large hackathon where the number of detected issues can be quite large. We propose a solution by adding a filter functionality to other columns of the issues table as well. The users will be able to filter issues based not only issue type but also other fields such as 'team name' and 'status'. Users will be able to set multiple value for each field, for instance, users will be able to filter issue of type 'Team needs Members' and 'Team needs help' at the same time.

Issue T5 The issue underpins that it is not readily easy for the user to comprehend what the 'message' field in the issue view means. Naming the field as 'message' is perhaps not intuitive enough for the users to understand what this field means. Following the solution the similar **Issue C1** where a better naming strategy will help alleviate this problem. We propose a better name for the 'message' field and change it to 'Relevant team Message'. In addition, we can introduce a little tooltip that will provide small explanation of what the field. In our case, upon hovering over the relevant entity field name, the tooltip will display that the field refers to the name of an entity, for e.g. 'The relevant text message sent by the team in Slack'.

Issue T6 Organisers want to know information on the current stage of the hackathon. This problem is relevant if the dashboard is meant to cater to multiple hackathons. However, our dashboard is meant to cater only for a single hackathon at a time. If the hackathon is using the dashboard for a single hackathon, the organiser is naturally aware of the stage of the hackathon. Hence, we can conclude that including information on the current stage on the hackathon is not necessary for our dashboard.

4.4.6 Advanced Proposed Features:

During our study interviews, experienced hackathon organisers came up with a host of ideas for expanding the existing functionality of the Dashboard. These proposed features did not directly related to the existing functionality. Here describe the ideas brought forth by the study participants. Later, we discuss on how we can approach the ideas including their validity, usability and implementation complexity in our current dashboard.

Proposed Ideas:

Users suggested on having a tickets assignment functionality. *"tickets assignment to mentors or organizers" - P2*. *"ideal to have collaborative tickets management between mentors and organisers" - P2*. *"I would like to assign some organizers/mentors on an issue" - P3*. *"... could have mentors assigned an issue directly through interface" - P2*. *"Team needs help issue auto-forwarded to relevant mentors through slack for example" - P2*. Such a functionality would allow organisers to assign a particular issue to a mentor. The benefits of having this functionality is three-fold. First, there will be less work for organisers of manually contacting a mentor and requesting him to help a team in case of 'Team needs help' issue. Second, this will allow for organiser to assign other types of issues to mentors as well and not only the 'Team needs help issue . This is especially relevant as in a large hackathon, the number of tickets can grow quite large and the few organisers may not be able to attend to the issue and they would instead like the mentors to chip in and help resolve the issues. Third, this will allow the organisers to assign an issue to another organiser who perhaps maybe the only one available or more capable for the issue at hand.

A sister functionality for the above functionality was proposed by the users to have a feature to let the other organisers or mentors know that they have been assigned an issue. *"...mentors are notified of the pinned tickets lined for them." - P3*. This functionality will allow for reminders or notifications of the issues assigned to the user when they log into the dashboard. Let us refer to these two proposed functionalities as **Issue R1**.

One of the users suggested to have a functionality that lets other organisers know that someone is working on ticket/issue. *"a way to let someone know if one is working on the tickets" - P2*. This will help avoid duplication of efforts and hence waste of organiser's time. While another user suggest to have a functionality that lets other organisers know when an issue has been resolved. *"letting other organizers know that an issue has been*

resolved" - P1. Going further, a user also suggested to have Slack based reminders to mentors for issues that they been assigned on *"I want slack based reminders to mentors" - P2*. This group of user concerns refers to organiser and mentor awareness of the issues/tickets. We can refer to these as **Issue R2**.

Another series of ideas by one of the user was related to providing functionality for communication between organisers on one side and mentors or participants/teams on the other. It was suggested to have functionality on the dashboard that allows at least sending emails to all the members of a team or alternatively having some contact info on the teams view. *"It is must have to being able to communicate with team members - fire emails or at least contact info" - P2*. The same user also suggested a way of sending group emails through the dashboard to mentors. *"group emails, emails to mentors through interface" - P2*. Let us refer to this as **Issue R3**.

An interesting suggestion was put forward by one of the users to add the ticket creation functionality to our dashboard. Such a functionality is expected to allow the participants/teams to create tickets/issues on the dashboard. This will allow for another way of ticket creation which is in addition to the existing auto-functioning issue creating backend. *"it is helpful for organizers to let teams create tickets on a platform rather than flooding DMs" - P1*. This can be an intuitive feature as a ticket directly created on dashboard is definitely shown to the organisers while an issue through chats may not be necessarily detected. Addition of this feature can complement our dashboard with feature creation and detection. Let us refer to this as **Issue R4**

Response to ideas:

Solution to the **Issue R1** asks for development effort for a couple of advanced requirements. Although implementing a requirement such as "auto-forwarding an issue to mentor through slack" is a little complicated to implement as we do not want an issue to be forwarded to all the mentors. We also do not want an issue to be forwarded to a mentor that may not be available. We can propose a functionality that allows tagging of mentors in issues by organisers. This will then fire a notification to mentors that a certain issue has been assigned to them. his will mean that we are designing a dashboard that is to be used by both mentors and organisers. However, for now we are focusing on dashboard that is specifically aimed for organisers. In case the dashboard does get to support the mentor role, this functionality could be implemented by allowing the

Organisers to tag mentors in an issue.

Issue R2 refers to organisers' awareness of other organiser's involvement in the same hackathon. Such a functionality requires us to develop a multi-user system where interactions of each organiser are captured by the system and then processed before providing this as awareness information to other organisers regarding the actions or any other engagement of the fellow organisers on the same hackathon. Slack based reminders to mentors for issues assignments also fall under the umbrella of awareness. However, there is development complexity associated with it. Organiser's awareness of the other's organisers was not our goal. Hence, supporting mentors and organisers for mutual awareness may be drifting far from our goal. Nonetheless, this can left for consideration in future iterations of the dashboard.

Although there maybe various ways to provide solutions for user's wishes in **Issue R3**, the design decision is going to be tradeoff between the development cost and functionality offered. Integrating communication platforms such as Slack or Email client into the dashboard pose large development effort and hence development cost. In contrast, an easy alternative to communication was mentioned by the user to provide with contact info for the team or the participant. This can implemented through existing slack data. Slack APIs provides with emails and slack username of the participants. Email can be used for writing an email while slack username can be used for finding the relevant participant(s) in Slack. The contact information can be presented to the organiser once the relevant team card or participant table entry is clicked upon. *"... being able contact team/mentor/participants by clicking on them" -P2*.

Issue R4 refers to letting participants create tickets/issues themselves on the dashboard for making organisers aware of any problems that they might be facing. The support for creating an issue can be provided through a form. The form may be the only part of the dashboard visible to participants. The participants can use a short form to create a ticket on the dashboard. Once a ticket is created, it is readily available to the hackathon organisers who can deal with it just they way they dealt with other tickets.

Issue	Comments	Response
Team Needs Members		

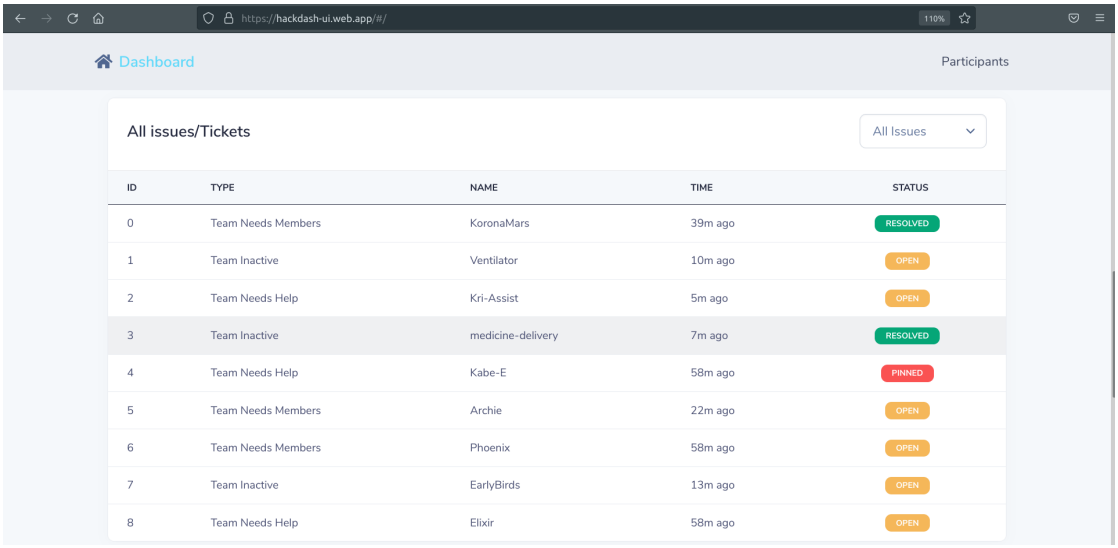
Issue	Comments	Response
N1	- not able to understand what 'joins' field means - some think 'joins' can mean people already in the team	- change name of the field to a more meaningful name such as 'new members'
N2	- 'joins' can be empty even if the members joined as auto-detect may not work always	- make 'joins' an editable field
N3	- cannot tell which members are in the team already	- to be addressed in issue C2
N4	- need feature to add participants to the team directly through dashboard	- issue does not relate directly to awareness - requires additional tech. infrastructure - may be a nice-to-have feature for future
N5	- need functionality to see the skillset of the participants	- skillset auto-detection is complex - participants can enter skillset as lastname in Slack (firstname will contain full name) - this lastname will then be used in the participants view for displaying skillset
Team Needs Help		
M1	- view mentors in a dedicated display that is for mentors only	- mentors information to be presented in a separate page
M2	- need information on skillset of mentors	- skillset auto-detect far-fetched functionality - organizers can manually enter skill-set of mentors

Issue	Comments	Response
M3	- need information on availability of mentors	- mentors can set the 'active'/ 'away' status in Slack - this will be read by Dashboard system to let organizers know of mentor availability
M4	- need information on timezones of mentors	- timezone information to be fetched from Slack user profile and shown in mentor view display
M5	- need functionality to filter mentors by different coulmns	- a table view of mentors with filterable columns
Organiser's Awareness of Teams		
C1	- confusing labelling of entity names such as 'name' for both team and participant name	- clearer labelling of fields such as 'team name' instead of just 'name' for teams and so on. - addition of descriptive tooltips to entity name fields
C2	- need to view the team structure (team name, participants, mentors) - need to have filter functionality in the teams view - display skillset of participants in a team's detail view	- teams view page to be added with teams table - get detailed team information upon clickingon a team from teams table - team filer functionality in teams table - related functionality implemented in related features such as - also resolved issue N3
C3	- need functionality to view and take action on all the issues relating to a team	- issues added as cards to a teams details view - issues can be acted upon just as in tickets table from the teams detail view as well
UI and Workflow of Tickets		

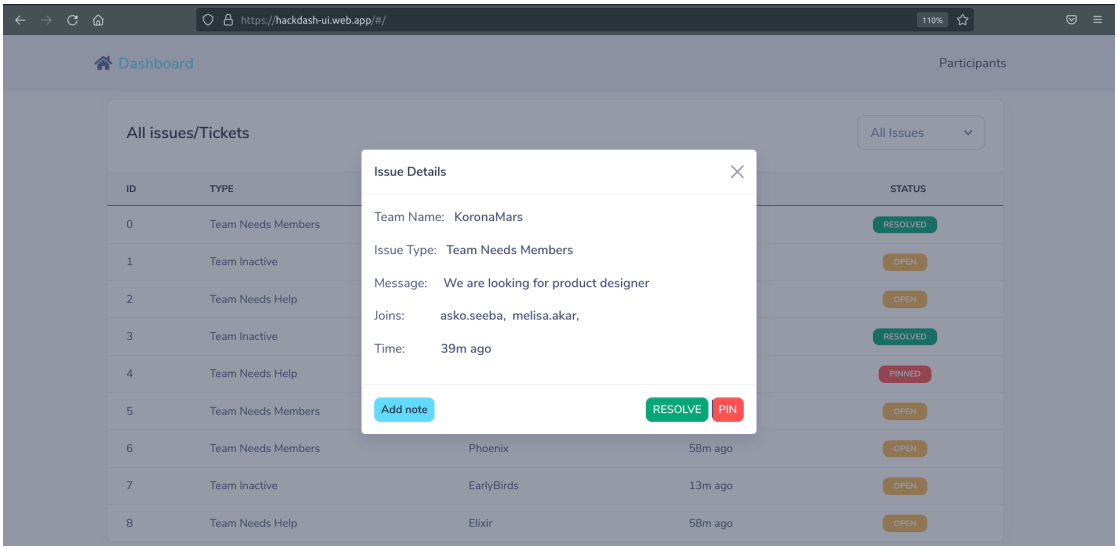
Issue	Comments	Response
T1	- need to know who added notes to an issue and when	- organiser's name to be added to notes - time stamp to be displayed for notes - notes split on separate lines, - notes sorted descending time
T2	- pinned issue should show a higher priority	- priority management for issues implemented - status wise and time wise composite priority
T3	- need functionality to set reminder for a given issue	- adding functionality to add reminder to an issue - reminder displayed as pop-up on dashboard
T4	- need functionality to filter issues in the issues table	- adding functionality to filter issues by multiple parameters in the issue table
T5	- unclear what the 'message' field in issues means	- renaming field to a more relevant name such as ' message from team' - adding a descriptive tooltip to the field
T6	- need information on current state of the hackathon	- irrelevant for our use case no further action
Advanced Proposed Features		
R1	- need a functionality to assign tickets to mentors directly from the dashboard	- technically complex, out of scope of awareness - no implementation now, maybe in future
R2	- functionality to let other organisers know that an issue has been resolved	- organiser's awareness of other organisers is not our goals, no further action for now

Issue	Comments	Response
R3	- functionality to ease communication b/w organisers mentors	- email address of mentors added to mentor detail
R4	- functionality for the teams to add tickets	- a dedicated page on dashboard for teams to create a ticket
Development of Signals		
S1	- team unhealthy communication may be a false alarm	- Slack/Discord can set as the medium of communication to avoid false alarm
S2	- teams are making use of voice messages for intra-team communication	- voice messages to be included in signal detection of type ’ unhealthy team communication’

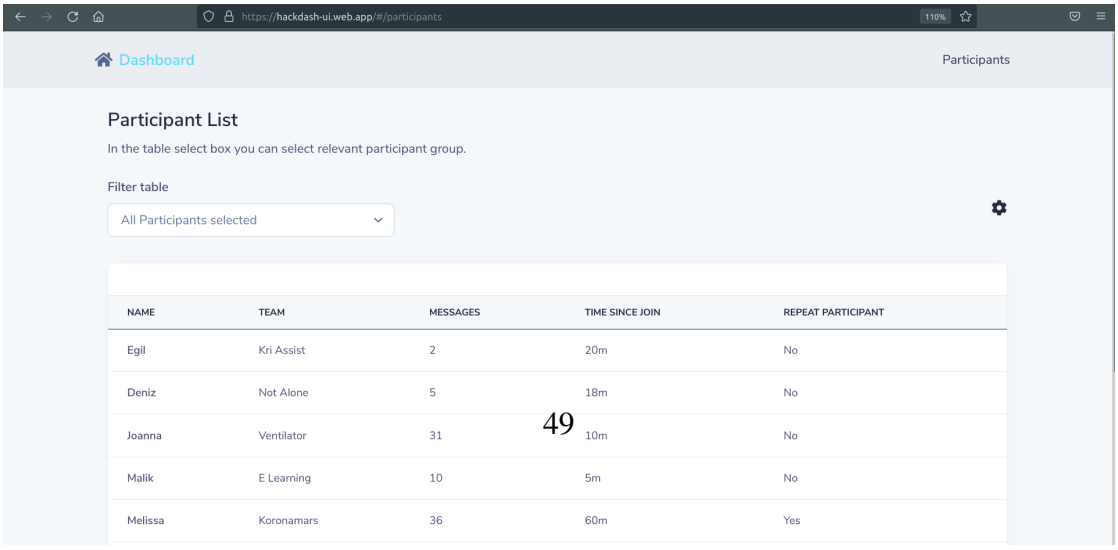
Table 4. Summary of Features from Study



(a) Issues Component View



(b) Issues Detail View



(c) Participant List View

Figure 4. Dashboard View

5 Discussion

5.1 Overview

As discussed in the section 2, there is literature support available on inter-team awareness. Yet, there was a gap in literature on how hackathon organisers can retain awareness of multiple teams. Through our findings in section 4.1 and further validation through study in section 4.4.1, we came up with three use cases as an answer to RQ1. These use cases are fundamental to organiser's awareness of online hackathon. The answer to this research question provides support on how can hackathon organisers retain an awareness of multiple teams. Our findings here expand on the existing inter-team awareness research in CSCW literature by adding support for awareness of teams for online hackathon organisers.

Answer to RQ1 provided us with an understanding of what information needs to be provided to hackathon organisers. However, this information does not exist and needs to be developed. For this purpose, we setup methods (described in 3.2) to extract necessary information from the relevant data available. We were able available to extract this information as presented in section 4.2 and then validate it though our study as demonstrated in section 4.4.1. Through this answer to RQ2, we further support our findings from RQ1 by demonstrating a validated process on how to extract the information pointed out in answer to RQ1. We established earlier in section 2 that awareness information needs to be complemented with relevant technology. Although, we do have necessary information, tools did not exist to present this information to the hackathon organisers. Through our findings presented in section 4.3, we were able to come up with dashboard solution to present the awareness information to hackathon organisers. We further validated our developed dashboard for usability through a study with experienced online hackathon organisers. The study validated the usability of the hackathon but also provided us with valuable insights into how the dashboard can be improved in future.

Through our work we are able to provide awareness to hackathon organisers in particular for the use cases in which teams may be needing help, teams that are looking for new members and any teams that may not be communicating actively over the communication platform. These use cases were not supported previously in theory or in

practise.

5.2 Proposed Improvements

There are certain improvements that we propose in order to make our findings more precise. Based on our methodology for answering the first research question in section 3.1 we were able to arrive at our findings as described in section 4.1. One of the things in the findings is the use case where organisers want to be aware of the unhealthy communication in teams. Through our evaluation study findings described in section 4.4, we can say that unhealthy intra-team communication is prone to several challenges. Although organisers did validate the usefulness of the features, it is clear that further work needs to be done in improving this feature. This can be done in several ways. One way is to have an alternative methodology for detecting unhealthy intra-team communication. While another approach is to use the feedback from the study as in section 4.4 for further improving the current methodology in constructing this type of signal. For improving our current methodology for developing signals, we need to eliminate the possible causes of false alarm for the unhealthy intra-team communication signal. These sources of false alarm are referred to as problems associate with signal detection in section 4.4.1.

Figure 5. MoSCoW Prioritization of Issues in Table 4

We employ MoSCoW framework [31] for proposing improvements to the dashboard. MoSCoW framework is well-known technique for prioritisation of features for iterative

development of software. These improvements are based on issues described in the section 4.4 and outlined in table 4. The table represents the issues that have to be dealt with in the next version of the dashboard according to the functionality proposed as response in table 4. Must haves represent the issues that are vital to functioning of the application. However, an additional must have not mentioned in the dashboard is need to have backend application for supporting the existing frontend dashboard application. Should haves represent features that will add convenience for the hackathon organisers but are not critical to the fundamental purpose of the dashboard. If the resources allow for, could have features will add further convenience for the hackathon organisers. Won't haves refer to the features which are perhaps out of our scope for now and can be definitely discarded or pushed to the backlog for future releases of our dashbaord.

5.3 Limitations

We formulated the research questions as follows:

RQ1: *What information do the hackathon organisers need in order to keep overview of a hackathon?*

RQ2: *Which information is available for keeping an overview of a hackathon?*

RQ3: *How can we present this information to aid organisers in keeping an overview of a hackathon?*

5.3.1 Limitations for RQ1

Through our methodology proposed in section 3.1, we made a theoretically founded use case selection. However, the use cases in section 4.1 do not cover all potential use cases of a dashboard for hackathon organizers. Consequently, we may not have covered all the potential areas in which a hackathon organiser may be needing awareness. This can perhaps also be deduced from the findings of our evaluation study where participants requested additional features. Although the features did not always fall under the context of organiser awareness, there were use cases that we failed to identify earlier in our findings of RQ1, for instance, organisers wanting to have an awareness of team structure.

We can say that the list of use cases we identified in section 4.1 may not be exhaustive and this can be a possible limitation of our work.

5.3.2 Limitations for RQ2

We had to make some compromises with the development of signals. We had limited access to participant chat data streams. We could not get measures for everyone was that some teams may not be using Slack for inter-team communication so we do not have sufficient information about them. Our signals are thus limited to providing suggestions based on the available data. Having all the teams in the analysis may have brought forth different perspectives.

5.3.3 Limitations for RQ3

For our evaluation study, our goal was to perform a qualitative evaluation and to identify means of improvement of our dashboard. In our study we made a founded selection of a few participants, however, their opinion may not be a good representation of hackathon organisers. Other hackathon organisers may have opinions different than those of participants in our study. The participants selected for study may not be representative of all hackathon organisers. Hence, our study may not have a fair mix of evaluation feedback. Consequently we may not have a fair evaluation of our dashboard.

In our evaluation, we presented the participants with a pre-drafted scenario. Then according to the scenario, participants were requested to perform a certain set of operations on the dashboard and asked specific questions according to the scenario. The focused nature of the questions may have limited the feedback from the hackathon organisers to very specific problems and we could have missed feedback that would have perhaps come up from a more open discussion.

We followed a thorough strategy for designing, carrying out and analysing the evaluation study results. The data gathered from study was analysed by a single person. This makes the analysis subject to interpretation bias. This bias is expected to influence the findings as well.

A fully-functioning dashboard is expected to have a supporting backend application in addition to the currently existing frontend [4]. The backend application is expected to

[4]https://hackdash-ui.web.app/

provide data persistence and integration with additional platforms such as Slack. Due to time constraints, we could not develop the backend. The dashboard in its current state was presented to the study participants for the evaluation study. The dashboard with a backend application could have behaved differently and led to a different experience for the study participants. This could have perhaps brought forth different findings from our evaluation study.

5.4 Implications

5.4.1 Research Implications

Although we did come up with three use cases for providing organisers with awareness of online hackathon, through our evaluation study there are additional aspects that organisers want to be aware of such as the structure of a team (addressed in section 4.4). This does not rule out the possibility there are not any other use cases, hence, we can say that more research can be done to identify additional use cases. The additional use cases can help us provide a better awareness of the online hackathon to the organisers. Accordingly, we can formulate a research question

RQ: *What additional cases do organisers need to be supported for providing them awareness of an online hackathon?*

5.4.2 Technical Implications

As mentioned earlier in this section of limitation, a fully functional version of dashboard application needs to have backend. A backend application will be integrated primarily with Slack Web API (Application Programming Interface). This integration will be providing our backend application with the required data pipeline. The backend application will perform processing analogous to the signal development methodology described in section 3.2 to detect issues with the teams and provide this information to the existing frontend dashboard application. Implementation of this backend application is the one of the next steps before implementation the dashboard in an online hackathon. Having the solution to must haves issues pointed out in MoSCoW prioritization in figure 5 is probably another of the next steps as well. Having solution to must haves will allow the hackathon organisers to use the dashboard in a online hackathon setting. Before

we can move onto the eventual deployment of the application on hosting platforms, the application needs to be made hackathon agnostic. This is a topic requiring expertise of DevOps practises. This design decision on how to make the application agnostic to a particular hackathon event will then influence the choice of hosting solution for the application. Although the exact design decisions require technical expertise in the domain of DevOps, we do know there are technical aspects yet to be dealt with.

5.4.3 Usage Implications

It is important to set-up tutorial for typical usage of the dashboard application. Following the general standards of documentation, the tutorial can a text and graphics based with clear illustrations and explanations. The tutorial can be based on a typical such as that presented to the evaluation study participants. The scenario can split into multiple sub-scenarios for focused needs of hackathons organisers in understanding the usage of dashboard. For easy accessibility, the tutorials can be hosted on the dashboard application instead of publishing on an external service.

Since, the dashboard has not been tested in an online hackathon setting, it will be safer to introduce it as a parallel system where hackathon organisers are not dependant on the dashboard for their awareness of the hackathon. Instead, the dashboard will be used on the side to test against its expected behaviour. It will be prudent to introduce the dashboard in an online hackathon following same structure similar to the hackathons H1-H3 presented in the table 1. Overtime, as organisers gain more trust in the system, the dashboard can be introduced as primary source of awareness for organisers and possibly in larger online hackathon settings.

References

[1] M. Komssi et al. “What are Hackathons for?” In: *IEEE Software* 32.05 (Sept. 2015), pp. 60–67. ISSN: 1937-4194. DOI: 10.1109/MS.2014.78.

[2] Nick Taylor and Loraine Clarke. “Everybody’s Hacking: Participation and the Mainstreaming of Hackathons”. English. In: *CHI 2018*. Conference on Human Factors in Computing Systems - Proceedings. Funding: EPSRC (EP/N005619/1); CHI 2018 ; Conference date: 21-04-2018 Through 26-04-2018. Association for Computing Machinery, Apr. 2018, pp. 1–2. ISBN: 9781450356206. DOI: 10.1145/3173574.3173746. URL: https://chi2018.acm.org/.

[3] Alexander Nolte et al. *How to organize a hackathon – A planning kit*. 2020. arXiv: 2008.08025 [cs.CY].

[4] Maria Angelica Medina Angarita and Alexander Nolte. “What Do We Know About Hackathon Outcomes and How to Support Them? – A Systematic Literature Review”. In: *Collaboration Technologies and Social Computing*. Ed. by Alexander Nolte et al. Cham: Springer International Publishing, 2020, pp. 50–64. ISBN: 978-3-030-58157-2.

[5] Alexander Nolte, Irene-Angelica Chounta, and James D. Herbsleb. “What Happens to All These Hackathon Projects? Identifying Factors to Promote Hackathon Project Continuation”. In: 4.CSCW2 (Oct. 2020). DOI: 10.1145/3415216. URL: https://doi.org/10.1145/3415216.

[6] Ei Pa Pa Pe Than et al. “Corporate Hackathons, How and Why? A Multiple Case Study of Motivation, Projects Proposal and Selection, Goal Setting, Coordination, and Outcomes”. In: *Human-Computer Interaction* (Apr. 2020). DOI: 10.1080/07370024.2020.1760869.

[7] Erik H. Trainer et al. “How to Hackathon: Socio-Technical Tradeoffs in Brief, Intensive Collocation”. In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work amp; Social Computing*. CSCW ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1118–1130. ISBN: 9781450335928. DOI: 10.1145/2818048.2819946. URL: https://doi.org/10.1145/2818048.2819946.

[8] HackerEarth Inc. *The complete guide to organizing a successful Hackathon*. https://hackathon-planning-kit.org/files/hackerearth.pdf. Accessed: 2021-04-07. 2017.

[9] Joshua Tauberer. *How to run a successful Hackathon*. https://hackathon.guide/. Accessed: 2021-04-07.

[10] MLH. *Hackathon Organizer Guide*. https://guide.mlh.io/. Accessed: 2021-04-07. 2020.

[11] Ei Pa Pa Pe-Than et al. "Designing Corporate Hackathons With a Purpose: The Future of Software Development". In: *IEEE Software* 36 (2019), pp. 15–22.

[12] DEVPOST. *Devpost*. https://devpost.com/. [Online; accessed 04-April-2021].

[13] DemocracyLab. *democracyLab*. https://www.democracylab.org/. [Online; accessed 04-April-2021].

[14] kreativdistrikt. *Kreativdistrikt*. https://www.kreativdistrikt.com/online-hackathon-platform/. [Online; accessed 07-April-2021].

[15] EvenTornado. *EvenTornado*. https://eventornado.com/. [Online; accessed 04-April-2021].

[16] TAIKAI. *TAIKAI*. https://taikai.network/en. [Online; accessed 07-April-2021].

[17] Tom Gross. "Supporting Effortless Coordination: 25 Years of Awareness Research". In: *Comput. Supported Coop. Work* 22.4–6 (Aug. 2013), pp. 425–474. ISSN: 0925-9724. DOI: 10.1007/s10606-013-9190-x. URL: https://doi.org/10.1007/s10606-013-9190-x.

[18] Carl Gutwin, Reagan Penner, and Kevin Schneider. "Group Awareness in Distributed Software Development". In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. CSCW '04. Chicago, Illinois, USA: Association for Computing Machinery, 2004, pp. 72–81. ISBN: 1581138105. DOI: 10.1145/1031607.1031621. URL: https://doi.org/10.1145/1031607.1031621.

[19] Douglas Zytko, Stephen Ricken, and Quentin Jones. “Group-Activity Organizing Through an Awareness-of-Others Interface”. In: CSCW ’18. Jersey City, NJ, USA: Association for Computing Machinery, 2018, pp. 361–364. ISBN: 9781450360180. DOI: 10.1145/3272973.3274097. URL: https://doi.org/10.1145/3272973.3274097.

[20] Gary M. Olson and Judith S. Olson. “Distance Matters”. In: 15.2 (Sept. 2000), pp. 139–178. ISSN: 0737-0024. DOI: 10.1207/S15327051HCI1523_4. URL: https://doi.org/10.1207/S15327051HCI1523_4.

[21] Google Inc. *Google Docs*. URL: https://docs.google.com/.

[22] Atlassian. *Confluence*. URL: https://www.atlassian.com/software/confluence.

[23] GitHub, Inc. *GitHub*. URL: https://github.com/.

[24] Alexander Nolte, Linda Bailey Hayden, and James D. Herbsleb. “How to Support Newcomers in Scientific Hackathons - An Action Research Study on Expert Mentoring”. In: *Proc. ACM Hum.-Comput. Interact.* 4.CSCW1 (May 2020). DOI: 10.1145/3392830. URL: https://doi.org/10.1145/3392830.

[25] Figma, Inc. *Figma*. URL: https://www.figma.com/.

[26] Jacob Nielsen. *10 Usability Heuristics for User Interface Design*. 1995. URL: https://www.nngroup.com/articles/ten-usability-heuristics/.

[27] Miro. *Online Whiteboard*. 2021. URL: https://miro.com/online-whiteboard/.

[28] Mark Handel and James D. Herbsleb. “What is Chat Doing in the Workplace?” In: CSCW ’02. New Orleans, Louisiana, USA: Association for Computing Machinery, 2002, pp. 1–10. ISBN: 1581135602. DOI: 10.1145/587078.587080. URL: https://doi.org/10.1145/587078.587080.

[29] Erin Bradner and Gloria Mark. “Why Distance Matters: Effects on Cooperation, Persuasion and Deception”. In: *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*. CSCW ’02. New Orleans, Louisiana, USA: Association for Computing Machinery, 2002, pp. 226–235. ISBN: 1581135602. DOI: 10.1145/587078.587110. URL: https://doi.org/10.1145/587078.587110.

[30] William Easley et al. “Understanding How Youth Employees Use Slack”. In: *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*. CSCW ’18. Jersey City, NJ, USA: Association for Computing Machinery, 2018, pp. 221–224. ISBN: 9781450360180. DOI: 10.1145/3272973.3274060. URL: https://doi.org/10.1145/3272973.3274060.

[31] *DSDM Agile Project Framework Handbook*. Agile Business Consortium, 2014. URL: https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Abdul Wahab**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Hackathon Organiser Awareness Dashboard,
   *(*title of thesis)

   supervised by Alexander Nolte, PhD
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Abdul Wahab
04/08/2021