UNILIFE - UNIVERSITY EVENT MANAGEMENT WEB APPLICATION REPORT

by

Paula Catalan

Group Information:

Group Name: Team 11

Submission Date: Friday 20th December 2024

Contents

Chapter 0: Introduction to Agile Development (SCRUM)

Agile development is a form of software development based on teamwork, planning in advance and incremental delivery. Its origins stem from the Agile Manifesto which was the work of seventeen software developers who met at a resort to discuss lightweight development back in 2001. Agile development's main values as defined in the Agile Manifesto are “Individuals and interactions over processes and tools”, which means people and their discourse are brought attention to over rigid processes and tools. Team collaboration to solve issues and challenges for just predefined systems.

“Working software over comprehensive documentation” is another core value of Agile development. Agile focuses on swiftly and quickly functional software and products over just constantly planning and excessive documentation. The point isn’t to undermine the importance of documentation, the primary goal is to deliver a working product which is far more important. “Customer collaboration over contract negotiation” is another value. Consistent collaboration with customers to tailor to their needs rather than trying to align strictly with a contract made early in development. This allows for a better product, ensuring it corresponds to a client’s real standards and anticipations. The final core value is “Responding to change over following a plan”. Change is a natural part of development, and this is something that Agile embraces. Being able to accommodate evolving conditions and demands by adapting plans will deliver enhanced results over attempting to fit everything into a rigid and static scheme.

The point of Agile isn’t to dismiss the importance and value in documentation, plans, contracts and processes, it's simply more effective to prioritizes communication, customer satisfaction and flexibility to deliver a better product efficiently.

Scrum was what was used throughout our group's development. Scrum is a type of Agile framework that focuses on flexibility and speed, using a series of versions and sprints to allow development to be agile.

Scrum is based on three pillars. Transparency, process needs to be visible to everyone, including the Scrum team and stakeholders. This transparency fosters a collective awareness of team activities, which is essential for effective collaboration and informed decision-making. This is done by having unambiguous and approachable Scrum artifacts, such as the Product backlog, Sprint backlog and Increment. Using a common language and clear definitions in Scrum is essential to prevent confusion. This practice guarantees that all team members and stakeholders share a shared understanding of terms, goals, and definitions. Without these shared definitions, miscommunication can become a problem, leading to misunderstandings.

Inspection, Scrum artifacts and progress must be inspected frequently to avoid deviations without interfering with the team’s work or progress. Regular examination of the work that’s currently in progress and present processes is necessary to make sure they align with the team's goals and the stakeholder projections. Although these inspections should be habitual, they shouldn’t interfere with team’s work or cause unneeded intrusion. Scrum events such as Sprint review, Daily Scrum and the Sprint Retrospective allow these inspections and checks to happen regularly without disrupting the team’s flow. The point of inspection is to find problems and issues early so they can be addressed without delay, assuring the team’s work stays in line with their Sprint and product goals.

Adaption, If something veers outside the plan, adjustments must be made in time and use an iterative and incremental approach to control risk and remain in line with their goals. The use of an iterative and incremental approach is urged by Scrum. Breaking done work into something more manageable and distributing them gradually Sprint by Sprint, teams may gain insight through each iteration, refine their process and continuously improve. The Sprint Retrospective, where team reflects on their execution and defines areas in need of improvement, each in relation to the product and the team's processes, is a crucial element of this process. Teams are motivated to consistently adapt and refine their processes to boost their effectiveness and efficiency.

The synergy of Transparency, Inspection, and Adaptation fosters a cycle of continuous gaining of knowledge and advancement. With these three pillars, Scrum teams can navigate uncertainty and complexity. Transparency helps teams make informed decisions and adapt quickly. Regular inspections keep teams aligned with goals, catch issues early, and make necessary adjustments. Adaptation ensures teams deliver valuable products and meet stakeholder needs.

A Scrum team is made from three key roles Product Owner, Scrum Master and Development Team with each playing their own distinct part in successful product delivery. Together, they form a self-organizing unit focused on achieving the goals of the project.

The Product Owner is the visionary of the product, responsible for defining what and why of the work being done. Their role is to create a clear product vision, ensuring it aligns with stakeholders' needs and business goals. The Product Owner maintains and prioritizes the Product Backlog making sure the most valuable tasks are tackled first and that priorities are clear to the team. They act as the bridge between stakeholders and the Scrum Team.

The Scrum Master leads and facilitates the team, ensuring they follow Scrum principles. They guide the team through the Scrum process, hosting daily Scrums, Sprint Reviews, and Retrospectives to keep work transparent and aligned. They remove obstacles, empower the team to work efficiently, and protect them from external disruptions.

The Development Team comprises professionals who are responsible for delivering the actual product increment and meeting the Sprint Goal They are a self-organizing group with all the skills necessary to design, develop, test, and deliver the work. The team collaborates daily to ensure progress is on track, manages the Sprint Backlog and focuses on delivering high-quality, usable work. The Development Team values collective ownership and accountability, working together to transform backlog items into functional deliverables while adapting to changes as needed.

Together, these three roles form a cohesive unit that thrives on transparency, inspection, and adaptation, ensuring the team delivers value incrementally and continuously while embracing change and collaboration.

Scrum has three main phases, with the initial phase being crucial for a successful project. This phase involves planning, setting objectives, and establishing the foundation to guide the team through development. The initial phase focuses on understanding the project's vision, led by the Product Owner who collaborates with stakeholders to outline goals, priorities, and value. During this phase, general objectives are set, the Product Backlog is created, and teams identify the most valuable features to prioritize. The planning process is collaborative and dynamic, ensuring alignment with stakeholder needs and the product vision

The execution phase is the core of Scrum, where development happens in time-boxed Sprints. It involves Sprint Planning to set goals and tasks, Daily Scrums to discuss progress and challenges, developing a "Done" Increment that meets quality standards, and Sprint Reviews to demonstrate the product increment and gather feedback from stakeholders. This iterative process ensures continuous improvement and alignment with stakeholder needs. Regular inspections and adaptations help the team stay on track and make necessary adjustments to deliver value effectively.

The closure phase in Scrum focuses on reflecting on team performance, identifying improvements, and officially completing the Sprint or project. Key activities include the Sprint Retrospective to review processes and identify actionable improvements, increment handover to finalize and release the product, and project closure to ensure all deliverables are met, lessons are learned, and team success is celebrated. This phase prepares the team for the next iteration and fosters a culture of continuous improvement.

In summary, the initial phase of Scrum involves defining the vision, setting objectives, and creating the Product Backlog, while setting up tools, processes, and team roles. The execution phase focuses on working in Sprints to deliver product increments, conducting Sprint Planning, Daily Scrums, and Sprint Reviews to inspect progress. The closure phase includes reflecting on team performance through the Sprint Retrospective, finalizing and delivering the product increment, and identifying areas for improvement.

Our group used Jira to handle and organize our tasks through the project. Jira helped us to create and allocate items in the backlog and measure progress during Sprints. We changed Sprint Leaders every one to two weeks, allowing everyone the opportunity to guide the team and manage workflow. We sustained consistent communication by holding an in-person Scrum meeting and an online Scrum meeting each week to ensure constant alignment. This combination of tools and practices helped refine our process and keep the team focused on achieving Sprint goals.

Scrum offers significant advantages, particularly in its flexibility and adaptability which make it ideal for projects with dynamic, evolving requirements. Scrum allows teams to respond quickly to changing priorities, plans, or stakeholder needs by using short, iterative Sprints This iterative approach enables regular feedback, helping teams align their work with the project vision while accommodating new insights or shifts in direction. Scrum's structure ensures that progress is made incrementally, reducing risks associated with large, upfront planning. Additionally, the frequent Scrum meetings like Daily Scrums, Sprint Reviews, and Retrospectives promote strong team communication, alignment with shared goals, and transparency in both progress and challenges. The emphasis on active and perpetual testing throughout each Sprint ensures that issues are detected early, leading to the delivery of a much higher-quality product compared to traditional development approaches.

Another key strength of Scrum lies in its focus on continuous improvement. The Sprint Retrospective, held at the end of each Sprint, provides an opportunity for the team to reflect on their processes, identify areas for improvement, and implement actionable changes in the next iteration. This cycle of inspection and adaptation allows both team and individual growth, promoting a culture of learning and refinement. By encouraging regular feedback loops and incremental delivery, Scrum helps teams remain motivated, engaged, and aligned with delivering value to stakeholders.

However, Scrum is not without its challenges. Its success heavily relies on the discipline, experience and collaboration of the team. Without these qualities, teams may struggle to follow Scrum practices effectively, leading to poor communication, missed goals, and stalled progress. A lack of accountability among team members can further exacerbate these issues, as Scrum's reliance on self-organization assumes that individuals will take ownership of their responsibilities. When team members fail to uphold this accountability, the entire project may suffer.

Additionally, Scrum is not always the most suitable methodology for every project. It thrives in complex, adaptive environments where requirements are uncertain and subject to change.

However, for projects that are simpler, linear, and well-defined, Scrum may introduce unnecessary overhead with its iterative structure and frequent meetings. In such cases, more traditional approaches may prove more efficient, as they emphasize sequential planning and execution without the need for constant reassessment. Thus, while Scrum excels in fostering flexibility, collaboration, and quality for complex projects, it requires careful consideration of team dynamics and project suitability to truly succeed.

In conclusion, Agile development, with its Scrum framework, provides an effective and adaptive approach to software development, particularly for dynamic and evolving projects. Grounded in the Agile Manifesto's core values, individuals and interactions, working software, customer collaboration, and responding to change Scrum embraces flexibility, collaboration, and continuous improvement to deliver high-value products. Its three foundational pillars transparency, inspection, and adaptation enable teams to make informed decisions, identify issues early, and adapt processes to achieve their goals. By working iteratively through Sprints, Scrum ensures progress is made incrementally, risks are minimized, and stakeholder feedback is consistently incorporated.

The structured roles within a Scrum team Product Owner, Scrum Master, and Development Team facilitate clear communication, guidance, and ownership, fostering a self-organizing and collaborative environment. Scrum's three phases of initial planning, execution and closure provide a roadmap for delivering products efficiently, encouraging teams to align with the vision, deliver high-quality increments, and continuously improve their processes.

While Scrum offers notable advantages like adaptability, enhanced communication, and improved product quality, its success depends on team discipline, accountability, and experience. For simpler, linear projects, Scrum may not always be the ideal choice. Nevertheless, when applied effectively, Scrum empowers teams to navigate complexity, deliver value consistently and foster a culture of learning and growth.

Chapter 1: Project Proposal

When we started our first meeting in week 1 as a team, we all shared with each other between one and two initial project ideas that we had brought with us and discussed them thoroughly. Most members had similar ideas that resembled some sort of student aid application, included me.

I brought a few underdeveloped ideas as back up, but the main idea I was able to present consisted of a personal planner app that was directed, primarily, towards students. Some of the main features it would include are:

- **To-do lists**: For day-to-day organization. The to-do lists would include timestamps or completion time goal. Ideally would also have an estimated time needed to complete the task. When the task is marked off as complete the item would be struck through to show completion.
- **Event calendar**: This feature would aim to help students be organized, meet deadlines more comfortably and avoid last-minute cramming. A wide view of the tasks that need to be submitted ahead, or nearing exams, would help students not miss any tasks that they might have to complete to meet the deadlines.
- **Space for annotation**: Note taking is essential to students so doing it in the same app would be very useful and save a lot of valuable time.
- **A pomodoro timer**: Pomodoro is a method of studying that consists of very intense 25-minute study sessions followed by a 5-minute break. This cycle is repeated 3 times. Then,

after one more 25-minute study session, the student takes a 15-minute break. Studies have shown that this method is one that avoids student exhaustion or unnecessary distraction, so it would be fitting to add it into the app.

My proposal was initially seen as favourable and was considered in the initial running. Other apps that were considered at the time were: Maynooth Marketplace, this app would simulate Facebook Marketplace but make it so that it was exclusive to Maynooth students, hopefully avoiding scams and keeping prices student friendly; New and improved event management app for Maynooth university, its students and societies, explained in more detail further down; And lastly an application that aided people to achieve their goals by keeping organized and receiving feedback from the app to ensure consistency in order to achieve said goal.

The idea of a Maynooth Marketplace was discarded as the ethical standards we wanted to maintain required a lot of authentications from users that seemed out of our reach skill wise. Furthermore, we couldn't agree on a way that the app would follow a zero-scam policy.

When it came to my app proposal I received quite a lot of feedback. It was a useful app and the scope of it was easy to grasp. It is something we all related to as users and felt like we could have quite a lot of expertise on. Moreover, the technologies needed were some we were either very familiar with or at least had enough resources to get acquainted with easily. However, when compared to the other proposed ideas, this one felt very flat. It was evidently a well thought out idea, but we could think of at least 3 or 4 examples of already working apps in the market that could do most if not all the features I was proposing. As a team we decided we wanted to innovate and make an app that might bring something new to the table.

We were finally down to two main ideas. An app that will help users reach their goals, and an event management app designed for universities.

The app to help users reach their goals was proposed as a mindfulness app that would help users stay focused and on track about their goals. The initial idea was for the app to pave the way for the user and create an action plan the user could use to reach their goal. However, questions were quickly raised by some of the team members as different goals can have very different action plans, how would we standardise the action plans? We quickly realized this wouldn't work, so ultimately redefined the idea as to making it exclusively a sports goal achiever app. However once again, the issue was raised as different sports require different, diets, training plans, number of resting days, etc. The scope of the project was too broad and would need to make use of some technologies that would've needed more time and attention, possibly an AI model.

The event management proposal consisted of an app that could be used by any university, but that for each version of the app only the university and its students and staff could access it. The university's societies and student union could use this platform to advertise events and manage the students that joined them. The venue and time could be updated, and users would be automatically notified. Students would also be able to create events and share them. They would have their own profile where they would have access to all events published on the portal. They could filter them by interest or event type. Furthermore, they would have access to all society profiles to decide whether or not they would want to follow them. This would also mean, societies would be able to keep track of the people who followed them, all in the one web application.

At first some of the team members were concerned the app would be too simple and that the scope of it wouldn't be particularly exciting. However, as we spoke about the app we started coming up with new implementations and ways the app could work that seemed to make the app bigger and better. Such implementations included: event cards that could give a glimpse of events so users can read through them quickly and only read in more detail the ones that actually are of interest to them. Additionally, we suggested that students should be able to follow each other as, some students might have really good reoccurring events that the user might be interested in attending, and many more.

Our chosen project proposal was the event management application. This app seemed the most manageable, skill and time wise. We were at least familiar with the technologies needed and all shared a very clear view of the potential scope of the project. Even though it took us a few hours of deep discussion we agreed upon this as our final idea.

Chapter 2: Technologies, architecture and testing

2.1 Introduction

The purpose of our project, UniLife, is to create an event management application designed specifically for university communities. The app aims to streamline how students, societies, and unions interact with events, providing an organized and user-friendly platform. With UniLife, users can register, log in, and access features like event creation, event registration, and following profiles of societies or unions.

This application proposal was chosen because it directly resonates with our experiences as students. We are best placed to "see" the problems and the needs of the university communities hence we were able to design a solution that addresses the issues tackled. Among other proposed ideas, UniLife stood out as the most feasible and achievable within our skillset and timeframe, while still offering opportunities to explore new technologies.

Key features of UniLife include user-friendly event creation tools for societies, event discovery and registration for students, and efficient sharing of event details and updates, making it a valuable tool for university life.

2.2 Technologies Used

We chose React.js, Firebase, and CSS for the front-end and back end of our application because they fit the needs of the platform and gave us a chance to improve our skills. While we had some experience with MongoDB, we decided to try Firebase Authentication and Firestore to learn something new and take on a challenge. React.js stood out because of its modern approach to building user interfaces and its widespread use in the industry, making it a great skill to pick up. Custom CSS allowed us to design the application exactly how we envisioned it, giving us full control over the look and feel.

React.js served as the backbone of our application, allowing us to create reusable components such as *NavBar* and *SearchBar*. Its component-based structure helped us break down complex features into smaller, more manageable parts. We used React's state management and useEffect hooks to dynamically display personalized content based on user account types (Student, Society, Union). For example, the *MyProfile* page rendered specific data, such as bios for students or descriptions for societies. Although React was new to us, the availability of detailed tutorials and documentation helped us quickly understand key concepts like state handling and the use of hooks such as useEffect (React Docs, 2024; LogRocket, 2024). One of the biggest challenges was managing asynchronous data fetching and ensuring components updated correctly when the underlying data changed.

Firebase was integral to data management in our application. Firestore queries were used to fetch user-specific information, events, and subscriptions, while Firebase Authentication handled secure logins and user sessions. We used Firebase's default local session persistence, which ensured users stayed signed in across sessions until they manually logged out. This provided a seamless user experience by reducing the need for frequent reauthentication (Firebase Docs, 2024).

We chose Firebase to explore a new technology, as our prior experience with MongoDB helped us understand database concepts but didn't include practical experience with authentication flows and their integration with databases. Implementing these functionalities was challenging at first, especially ensuring data integrity, such as saving edits to profiles and associating events with the correct hosts. However, Firebase's tutorials on Firestore and Authentication helped us overcome these challenges and optimize our implementation (Firebase Docs, 2024).

For styling, custom CSS allowed us to create a visually appealing and responsive interface. It gave us full control over design elements such as background gradients, profile sections, and event cards.

The main challenges we faced included managing asynchronous data fetching, implementing logic for different account types, and ensuring data was correctly stored and retrieved in Firestore. Despite these difficulties, React's modular structure, Firebase's robust tools, and custom CSS allowed us to build a dynamic and user-friendly application.

2.3 Architecture

Our website is based on a client-server model where the server stores all the event and user information, and the user uses the interface which interacts with it.

Front end displays the user interface, interaction and talks with the backend to push and pull data from the database. It is developed using React making it a Single Page Application (SPA). Backend was built with node.js and express, but mostly by the use of built in Firebase functions. It included the handling of requests, error management and authentication. We used Google's firebase to host our database.

We used Google Cloud to host our site. The code was cloned directly from our GitLab "main" branch repository. Click the link below to deploy our site (available until Feb 2025).

http://35.214.114.190:8080/

We used React.js and CSS on the front end to create a dynamic and clean user interface, while Firebase handled all the back end related things we needed such as the database authentication. The front end interacts with Firebase through its APIs, so when users do things like creating/joining events, creating/joining societies or updating their profiles, the data is sent to Firebase instantly, updated in the database, and reflected in real-time in the app. Logins are a breeze too, since Firebase Authentication is capable of handling all of that with no issue. Everything is also built with a modular approach in mind—React components like Header, Footer, or smaller ones like User Card are reusable, so adding features or fixing bugs is way easier. CSS ties it all together with consistent styling that works well overall. The user flow is also straightforward: log in, do whatever task you were aiming to do (joining, creating events etc.), and Firebase takes care of saving it all to the proper database it belongs to.

The backend structuring consists of five different collections:

1. Users: contains key personal details of each user, the societies and/or students they follow, account type and societyId or name depending on account type.
2. Student number: links student number to user id.
3. Events: key information of event such as date, location, descriptions, image, and the host's email address.
4. Societies: an extension of user, contains only society users and their information.

5. JoinedEvents: links user's email to an array of eventIds.

As users, depending on their account type they can take different paths through the application. All users can log into the application and will be redirected to their home page. They will all be able to access their profiles (MyProfile) where they can edit their profile and create events. All users can also view all societies in a grid-like manner by clicking "Societies". Lastly, they can all Log Out. Students have a few more options including "MyFeed" which only shows events from the societies and students the user follows and "MyTickets". After purchasing tickets, the student will be able to view them under said page.

Find UML diagrams in appendix Figure 2.1, Figure 2.2 and Figure 2.3.

2.4 Testing

Frontend Testing

Frontend testing for our app primarily involved manual testing to ensure usability, responsiveness, and cross-browser compatibility. During the build process, we used a rather iterative testing approach, we tested the user interface by interacting with key features, such as the profile page, event creation workflows, and the subscription functionality. For instance, while developing the myProfile page, we tested the different edit profile functionalities since we had multiple user types and each account type had different edit profile features. This continuous testing allowed us to identify issues early and kept up the workflows momentum. To validate that buttons and interactions worked as expected, we relied on console logs, such as console.log("Follow button pressed"), to confirm user actions were being captured correctly. This helped us identify and resolve UI inconsistencies and bugs in real time.

We made an effort to design the application to be as responsive as possible within the time available. Testing across various device resolutions, including desktops, tablets, and smartphones, allowed us to ensure that the app adapted reasonably well to different screen sizes. Cross-browser testing was conducted on popular browsers like Chrome, Firefox, and Edge to verify consistent layout and performance.

A major focus of testing was linking the frontend with Firebase. We used console logs to monitor which user data was being fetched, ensuring that profile and event data were displayed accurately and updated dynamically. One significant challenge was conditionally displaying UI elements and implementing button interactions based on the user's account type. Through careful debugging and iterative improvements, we ensured that buttons and other interactive elements, such as following profiles or saving edits, performed their intended functions correctly.

Backend testing

The most important objective of our app's backend testing was to make sure that the Firebase database worked properly. We spent a while making sure that Firestore could properly read, write, and update the data for the societies, user, and events collections. We made use of console logs, like console.log ("Data saved successfully") or console.log ("Fetching data..."), to keep track of the information flow in order to achieve this. This helped us to make sure that all the data that was being written on the database was handled correctly. Testing real-time updates was one of the harder parts. For example, when a user updated their profile or created a new event, we had to make sure those changes showed up to other users when they, for instance, refreshed the events page. In order to do this, we had to keep an eye on the database/firebase logs and ensure the change was also translated into the front-end area of things in order to verify it all worked properly. We also conducted backend validation tests to ensure data integrity. On the registration form we implemented logic that only allows users to register with a valid university email address (ending @mumail.ie), unique student number or society ID. To test this we used invalid emails and student numbers/ID's that were already taken. Backend logs were helpful here again, console.log("Invalid email address") or console.log("Invalid Student Number") were used to help us confirm that the system correctly rejected invalid entries.

2.5 Lessons Learned

Challenges

While developing our application we came across several challenges in both front-end and backend aspects. One of those challenges was integrating Firebase with our React frontend. Setting up Firestore queries to fetch user-specific data, such as profiles, events, and subscriptions, required careful management of asynchronous operations. Debugging issues like missing or incorrect data often disrupted the flow of the application. We addressed these issues by implementing clear loading states and using console logs to track data retrieval and ensure accurate rendering.

Another challenge was implementing dynamic functionality for different account types (Student, Society, Union). This required designing the UI logic to conditionally display data and features, such as profile editing or event creation, based on account type and ownership. The backend played a supporting role by providing the necessary data, such as the accountType and user details, which the frontend used to determine what features should be accessible. Errors in Firestore queries or account type handling occasionally caused unintended behaviors, such as incorrect or incomplete data being fetched and displayed.

Cross-team collaboration added complexity, especially when linking the frontend with Firestore. Ensuring data integrity during updates, like saving edits to profiles or associating events with hosts was critical but challenging, as incorrect updates could cause inconsistencies.

Skills Gained

During this project we gained a lot of practical skills from using all these different tools to build a real application. Using React.js for example, we were able to learn how to effectively use hooks like useEffect to better manage component-based structures and dynamically display content to the users. The implementation of user authentication, which was necessary for the application, and the correct ways to effectively use Firestore and take advantage of all its features were crucial skills that we were also able to pick up from familiarizing ourselves with firebase which was new to all of us. In terms of CSS, this project helped us produce more responsive designs and improve our ability to offer users a better experience. All things considered, the hands-on experience we obtained from debugging and fixing different problems that came across while building the application definitely helped us with our problem-solving skills too.

Chapter 3: The SCRUM process

3.1 Initial scrum meeting

The first Scrum meeting was an essential starting point, laying the foundation for our project. Each participant introduced themselves, shared their coding experience, and expressed their preferences for working on the frontend, backend, or both. These discussions ensured a fair distribution of roles while considering the strengths and interests of each developer.

Nathan, Paula, and Anshuk focused on frontend development because of our interest in creating an interactive user interface. Dmitrij showcased flexibility by contributing to both frontend and backend tasks, effectively bridging the two. Meanwhile, Sam, Patrick, and Chikezie dedicated their efforts to backend development, concentrating on database management and business logic.

Once roles were assigned, we brainstormed potential project ideas, including a student marketplace, a digital planner, and a sleep tracker app. After evaluating these options for feasibility, complexity, and relevance, we decided on the Uni Life project, a campus event management app. This project aligned perfectly with our target audience of students, societies, and student unions and was achievable within our time constraints. The team was excited by the project's meaningful goals and its technically challenging scope.

In the same meeting, we finalized our tools and technologies. React and JavaScript were selected for their flexibility and widespread use in web application development. Firebase became our backend platform for managing data and user authentication. Jira was chosen for task management, helping us structure sprints, track progress, and monitor tasks effectively. We planned weekly sprints to ensure consistent progress and held regular meetings to address challenges and refine our processes.

This meeting provided a clear direction for the project, aligned our goals, and established a framework for effective collaboration.

3.2 Project estimation

Accurate task estimation was a cornerstone of our project's planning process. Breaking tasks into smaller, manageable activities and assigning story points based on complexity allowed us to distribute workloads effectively across the team. Straightforward tasks, like creating static forms for user input, were assigned 3 points. Moderate tasks, such as adding form validation or integrating frontend elements with backend APIs, received 5 points. More complex tasks, requiring significant time and collaboration like implementing an event filtering algorithm were assigned 8 points or more.

Frontend tasks included designing the registration survey, creating the event creation page, and implementing a personalized "My Feed" feature. On the backend, tasks like role-based access control, database schema design, and event filtering logic were key.

To determine story points, we used Planning Poker, a collaborative estimation technique that allowed team members to discuss and agree on task complexities. This ensured transparency and credibility in our estimates.

As the project progressed, we refined our estimates to reflect evolving challenges and new insights. For example, the event filtering feature required more backend logic than anticipated, leading to an increase in its story points from 5 to 8. Similarly, additional time was allocated for making the "My Tickets" page responsive across different devices.

Jira played a central role in organizing the project. It provided visibility into task priorities, progress, and dependencies, enabling us to reallocate workloads as needed. By structuring tasks into weekly sprints and adjusting priorities dynamically, we avoided overloading team members and maintained consistent momentum. This process highlighted the importance of flexibility and adaptation in maintaining quality while meeting deadlines.

3.3 Meeting Minutes

Regular and Online team meetings were vital for tracking progress, addressing challenges, and planning next steps. These sessions were held both virtually and in person, depending on convenience, and fostered effective communication and collaboration.

During the initial meetings, we focused on completing tasks for the early stages of the project. In an October 30 virtual meeting, we discussed frontend tasks for the user survey and event creation pages, while the backend team worked on society collection and survey population logic. We also strategized on integrating Firebase Cloud Messaging for notifications, an innovative feature to improve user engagement.

By November 6, we had completed the second sprint. This meeting allowed us to review the functionality of the user registration process and address layout adjustments for the homepage. These discussions clarified individual responsibilities and identified potential roadblocks.

As the project progressed, our meetings delved deeper into integration and testing. On November 20, we tested and integrated backend routes for event creation and survey submissions, ensuring a smooth workflow. We also customized user experiences for students, societies, and student unions, which required close coordination between the frontend and backend teams. These tailored experiences addressed the unique needs of each user group, a critical success factor for the app.

The November 27 meeting focused on completing the "My Tickets" page and finalizing CRUD operations for events. We also began mapping the app’s session flow to ensure smooth navigation and continuity. Future enhancements, such as improving the user experience and adding a notification system, were discussed and prioritized.

Our final meeting on December 11 served as a wrap-up for the project. We reviewed the app’s functionality, integrated dynamic event displays on the Home and My Feed pages, and improved the maintainability of the codebase. We also planned for deployment and finalized project documentation. These discussions ensured alignment with our objectives, addressed remaining challenges, and enhanced the app’s overall quality.

Systematic scheduling and effective minute-taking ensured that each meeting contributed to the project’s momentum. These discussions helped us overcome obstacles and identify opportunities for improvement, resulting in a cohesive and functional application. Reflecting on the project, we are confident that consistent communication and teamwork were instrumental in our success.

3.4 Burndown Chart

The burn down chart was an essential tool for tracking sprint progress and ensuring timely task completion. This chart visualized the remaining workload versus the sprint timeline, helping the team identify whether they were on track. Figure 3.1.

Implementation

The burndown chart was updated regularly, reflecting task completion rates in Jira. Tasks were marked 'To-Do,' 'In Progress,' 'Code Review,' 'Testing,' and 'Done’ ensuring transparency. For instance:

- During the first sprint, tasks like user authentication and homepage wireframes were marked as complete, demonstrating steady progress.
- In subsequent sprints, tasks related to event creation and RSVP systems were tracked, with a noticeable dip in the workload as these features were implemented.
- And so on until the end of Sprint 5.

Burndown Chart

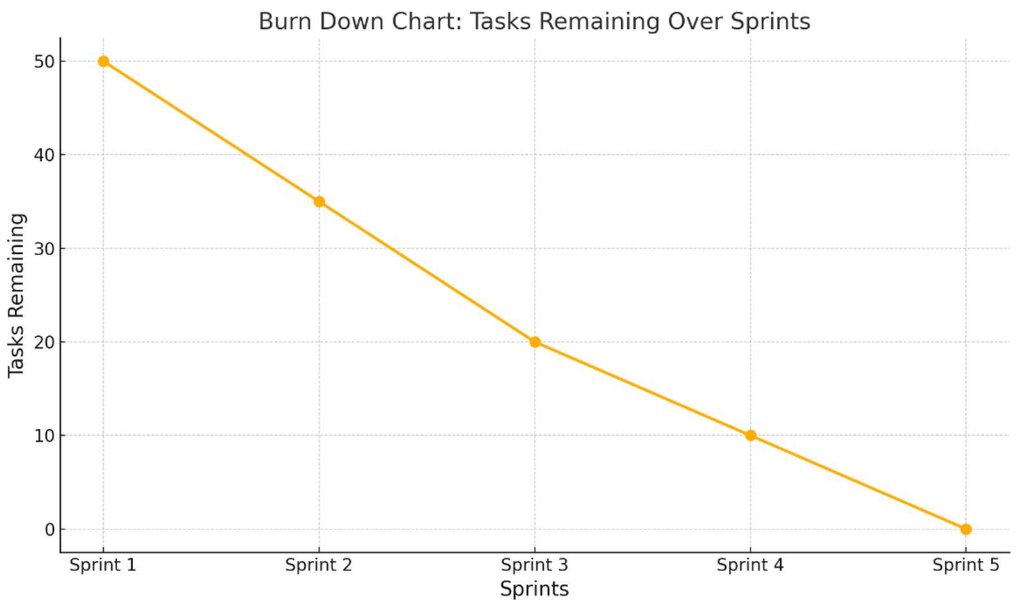
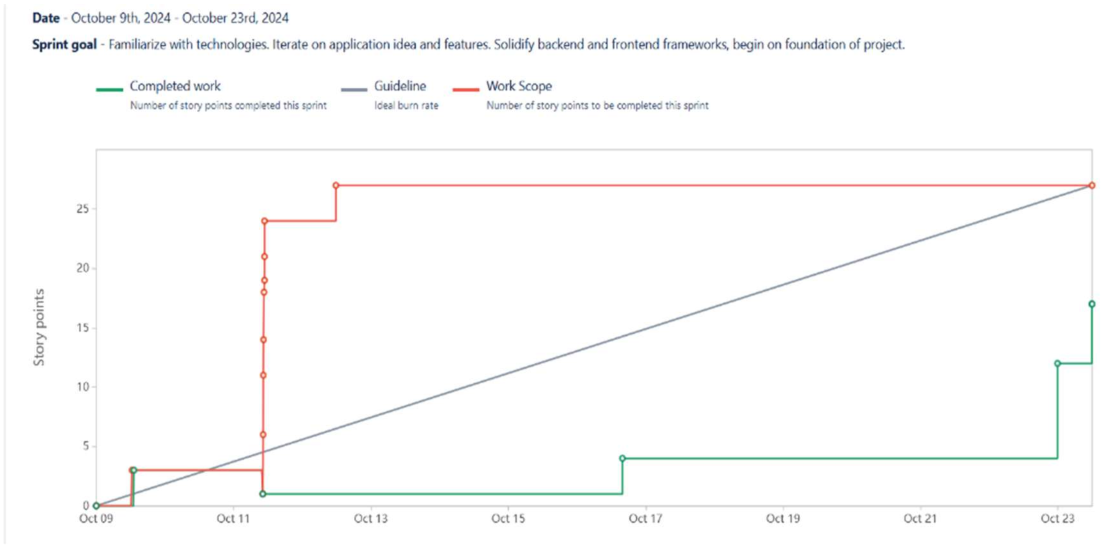


Figure 3.1: Burndown Chart of full Epic

For completion purposes we also wanted to include some further graphs related to the project progress.

Burnup Charts

The burnup chart includes the Completed work line (Green) and the Work scope line (Orange). Jira also includes a guideline (Grey) that would be the ideal burn rate of task completion. See the following graphs for each individual sprint.


Date - October 9th, 2024 - October 23rd, 2024

Sprint goal - Familiarize with technologies. Iterate on application idea and features. Solidify backend and frontend frameworks, begin on foundation of project.

Figure 3.2: Sprint 1 Burn up Chart

Date - October 23rd, 2024 - November 6th, 2024

Sprint goal - The goal is to have functioning backend components that update upon registration. A survey for students to fill out. To create a User Interface for the user homepage. If possible, start on the backend for this page. Brainstorm Notifications.

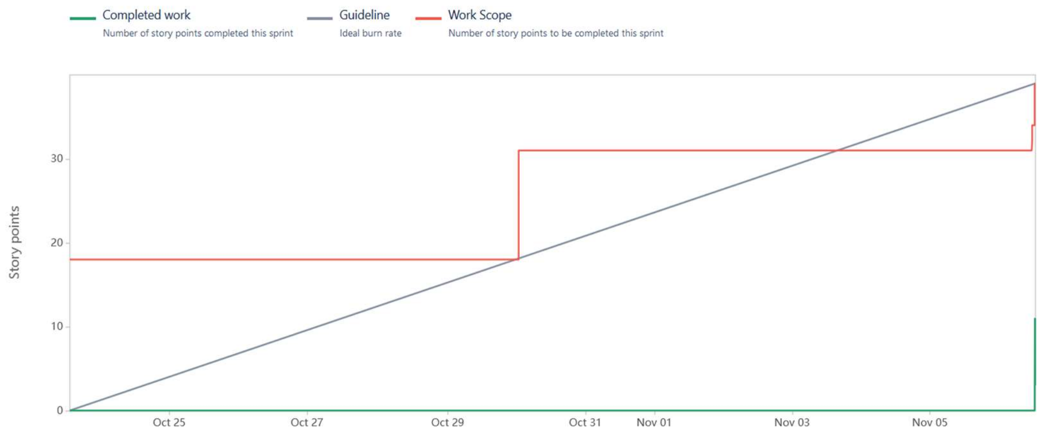


Figure 3.3: Sprint 2 Burn up Chart

Date - November 6th, 2024 - November 20th, 2024

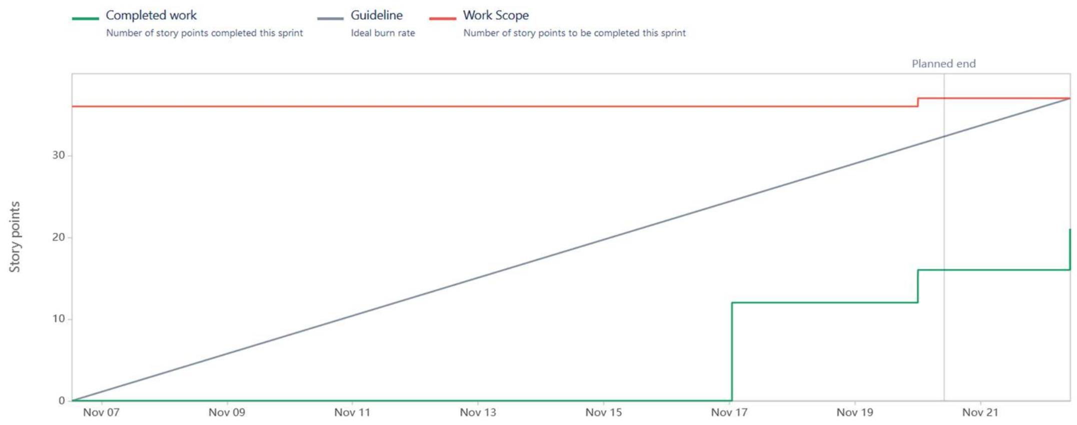
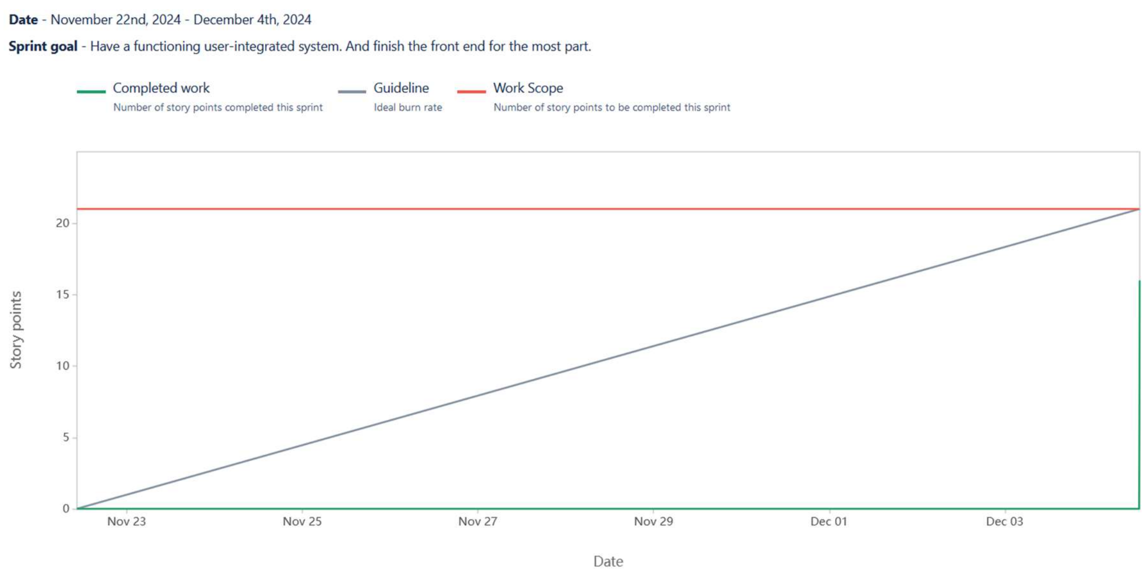


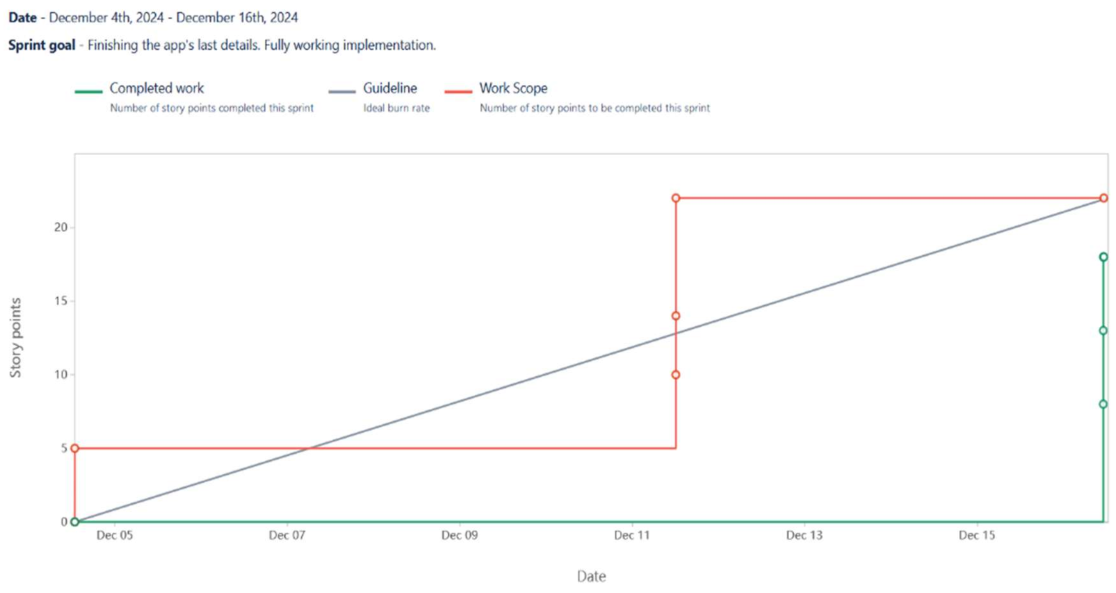
Figure 3.4: Sprint 3 Burn up Chart



Date - November 22nd, 2024 - December 4th, 2024

Sprint goal - Have a functioning user-integrated system. And finish the front end for the most part.

Figure 3.5: Sprint 4 Burn up Chart



Date - December 4th, 2024 - December 16th, 2024

Sprint goal - Finishing the app's last details. Fully working implementation.

Figure 3.6: Sprint 5 Burn up Chart

Cumulative Flow Diagram

A Cumulative Flow Diagram (CFD) is a way to visually monitor the flow of work across different sprints regarding the workflow over time during the development project. It mostly focuses on the team's efficiency and shows if there were any bottlenecks due to roadblocks. Figure 3.7 shows that as time progressed, we completed more and more tasks and that the number of 'To-do' tasks and ones that were 'In Progress' decreased quite linearly.

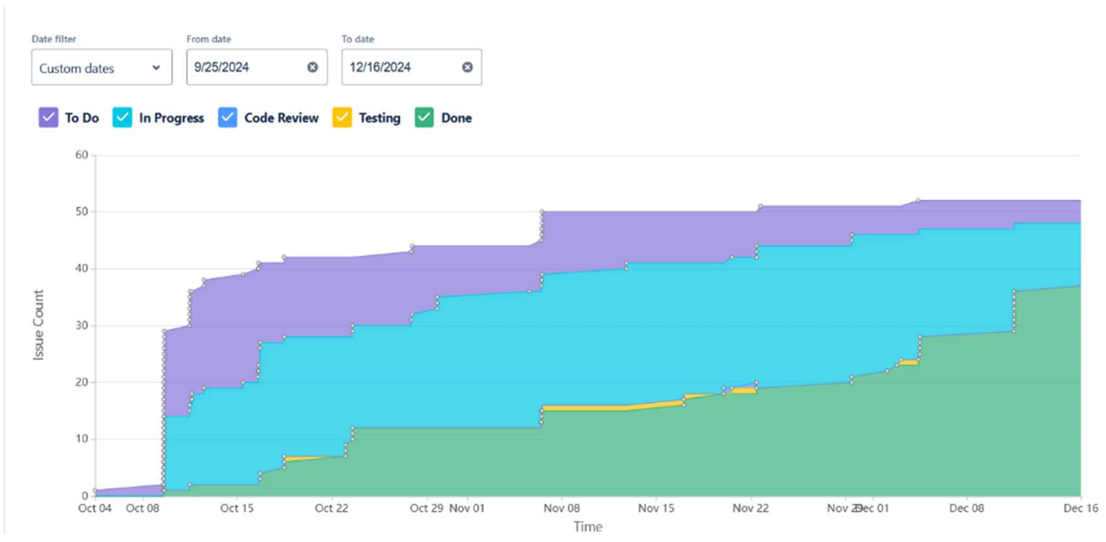


Figure 3.7: Cumulative Flow Diagram for Team 11

Velocity Report

The Velocity report shows how much work was done during the sprint. This is compared against the work amount of work in the sprint when it begun. Our report shows that with the exception of a very productive first sprint, we gradually increased our amount of completed work. See Figure 3.8.

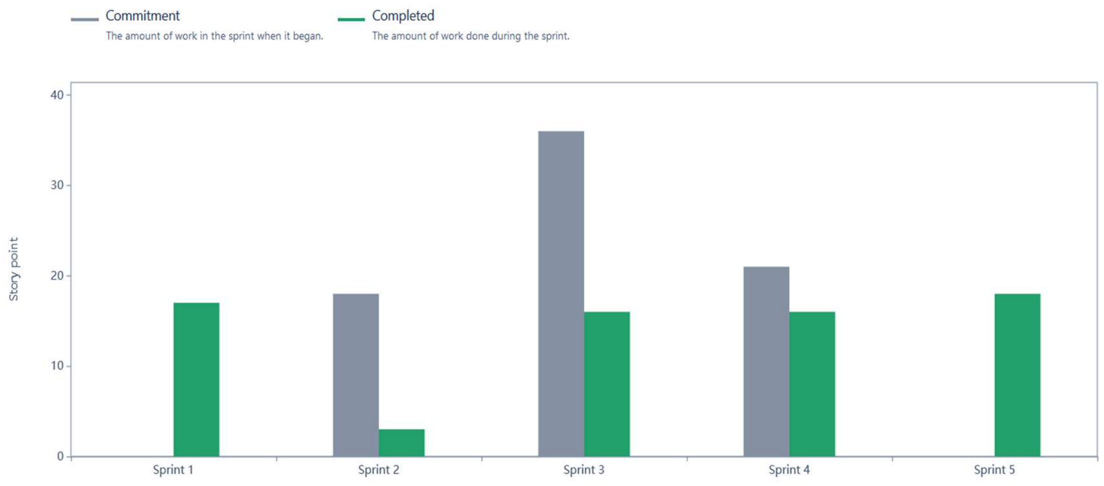


Figure 3.8: Velocity Report for Team 11

3.5 Story Maps

We decided to use a software application named Miro compiling the user stories into a user story map. Miro is a digital collaboration platform designed to facilitate remote and distributed team communication and project management. We took advantage of the advanced mapping capabilities that Miro offers to create our map.

The user stories went through a lot of change since the beginning of the project. There were some user stories that we simply did not see were worth incorporating into our project based on how complex they would be to implement. This was a wise decision as looking back now from the end of sprint four, we would have struggled to finish the application, the sacrifice for time would have not been worth it. Some of these stories include…

1. As a society leader, I want to monitor RSVPs so that I can adjust event resources and spaces accordingly.
2. As a student, I want to RSVP to events so that I can secure a spot and receive event updates.
3. As a student, I want to provide feedback on events I attended so that organizers can improve future events.
4. As a student, I want to vote on proposed events so that I can have a say in which events are organized.

Regarding to the RSVPs, we decided that a confirmation feature may be too difficult to develop under our tight time constraint, in an ideal world this would have been a great feature however it was not necessary therefore we left it to the side. The feedback and voting polls would have required a forum-like page and a lot of backend development with restrictions and privileges so we decided not to implement it.

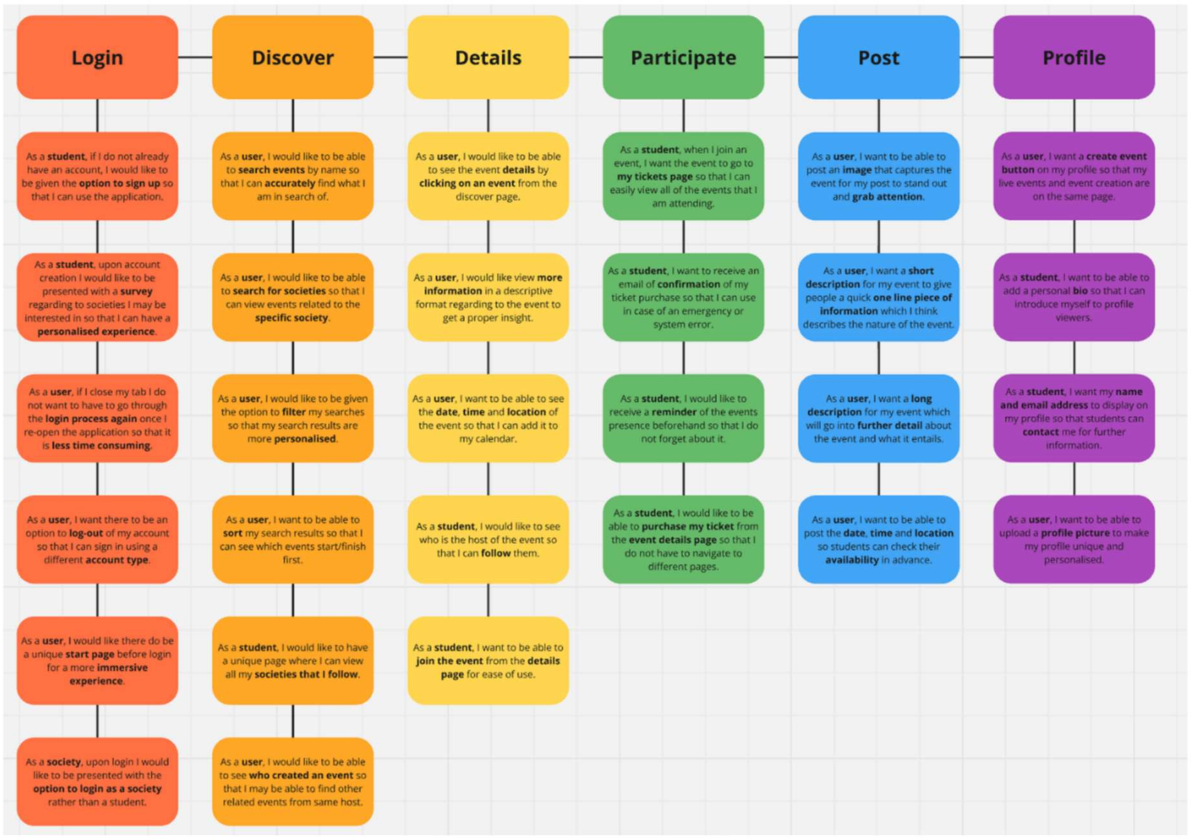
On the other hand, there were stories we were planning to implement but never got the chance to because we ran out of time, I will talk more about these stories in “3.7 Future releases/sprints”.

A tool that really helped with putting the map together was the Jira board. We created an issue type named “Story”, this acted as a group or collection of stories. The reason why it was so helpful is because Jira offers a timeline feature so we could visualise how the stories adapted to the application over numerous sprints. Not only did it offer a timeline feature but also a checklist so once a story was incorporated, we could check it off the list, this being the reason why I could easily pin-point which stories we decided not to include and which ones we never got a chance to develop.

We thought of a total of six User Goals being…

1. Login: The login/sign-up page and the introductory page.
2. Discover: The home page and browsing.
3. Details: Event specific details.
4. Participate: The process of joining an event.
5. Post: Creating an event.
6. Profile: Individual user profiles.

The twenty-nine steps can be seen in the User Story Map below…



Login	Discover	Details	Participate	Post	Profile
As a **student**, if I do not already have an account, I would like to be given the **option to sign up** so that I can use the application.	As a **user**, I would like to be able to **search events** by name so that I can **accurately** find what I am in search of.	As a **user**, I would like to be able to see the event **details** by **clicking on an event** from the discover page.	As a **student**, when I join an event, I want the event to go to **my tickets page** so that I can easily view all of the events that I am attending.	As a **user**, I want to be able to post an **image** that captures the event for my post to stand out and **grab attention**.	As a **user**, I want a **create event button** on my profile so that my live events and event creation are on the same page.
As a **student**, upon account creation I would like to be presented with a **survey** regarding to societies I may be interested in so that I can have a **personalised experience**.	As a **user**, I would like to be able to **search for societies** so that I can view events related to the **specific society**.	As a **user**, I would like view **more information** in a descriptive format regarding to the event to get a proper insight.	As a **student**, I want to receive an email of **confirmation** of my ticket purchase so that I can use in case of an emergency or system error.	As a **user**, I want a **short description** for my event to give people a quick **one line piece of information** which I think describes the nature of the event.	As a **student**, I want to be able to add a personal **bio** so that I can introduce myself to profile viewers.
As a **user**, if I close my tab I do not want to have to go through the **login process again** once I re-open the application so that it is **less time consuming**.	As a **user**, I would like to be given the option to **filter** my searches so that my search results are more **personalised**.	As a **user**, I want to be able to see the **date**, **time** and **location** of the event so that I can add it to my calendar.	As a **student**, I would like to receive a **reminder** of the events presence beforehand so that I do not forget about it.	As a **user**, I want a **long description** for my event which will go into **further detail** about the event and what it entails.	As a **student**, I want my **name and email address** to display on my profile so that students can **contact** me for further information.
As a **user**, I want there to be an option to **log-out** of my account so that I can sign in using a different **account type**.	As a **user**, I want to be able to **sort** my search results so that I can see which events start/finish first.	As a **student**, I would like to see who is the host of the event so that I can **follow** them.	As a **student**, I would like to be able to **purchase my ticket** from the **event details page** so that I do not have to navigate to different pages.	As a **user**, I want to be able to post the **date**, **time** and **location** so students can check their **availability** in advance.	As a **user**, I want to be able to upload a **profile picture** to make my profile unique and personalised.
As a **user**, I would like there do be a unique **start page** before login for a more **immersive experience**.	As a **student**, I would like to have a unique page where I can view all my **societies that I follow**.	As a **student**, I want to be able to **join the event** from the **details page** for ease of use.			
As a **society**, upon login I would like to be presented with the **option to login as a society** rather than a student.	As a **user**, I would like to be able to see **who created an event** so that I may be able to find other related events from same host.				

Figure 3.9: Story Map Start of Sprint 1

3.6 Testing

Testing in our project was not only seen from a technical perspective, verifying inputs and outputs but also meant we as a team had to ensure excellent communication in the form of open dialog and frequent updates. The testing stage didn't really start picking up until after the first few weeks of coding. At the time we were mostly focusing on the design, redesign and the very beginning stages of our coding journey for this project. It was essential to keep testing in an agile way, as code was being modified and implemented consistently throughout the development process. What worked at the start might have been affected by the new incoming code. Hence, it was imperative that we ensured testing was carried out the whole way through our development process.

Testing at first included and was mainly focused on the testing of individual components by the developers that created them. In doing this, we ensured that the integration of new features wouldn't disrupt the flow of the application as well as avoiding any errors cascading into more concerning system-wide issues. As we integrated all components into one system, all team members would log into the app and test different test data ensuring it was fully functioning from all perspectives. Full system testing started being done around the halfway mark of our third sprint, when a considerable number of features were integrated with one another within the application.

Half of our development team focused on front end, meanwhile the other half focused on the backend development. The division of tasks allowed us to not only work faster and more efficiently, but also to create some unwritten handoff rules that needed to be met before another developer could jump in and add their section of code. Before code was sent from the front end to the backend development team, it was necessary to ensure proper functionality. The iterative nature of scrum actions a constant streaming of trial-and-error testing given the continuous integration of new code.

We found it is imperative that data is tested on its edge cases and forced errors as allowing an unseen error to go undetected for long, increases the difficulty of debugging further down the line. Thanks to our continuous stream of testing, we believe we saved heaps of time, which we were able to dedicate to further development of more features.

Given this, we closely worked together in pairs for the project. We would corporately build our programs to align with each other's needs a preferences thinking ahead of time. We feel comfortable we have built a robust and mostly error free piece of software through collaborative work and iterative and inclusive testing.

3.7 Future Releases

There are many features and functionalities that we never got an opportunity to implement into the application due to time constraints and other complexities. If we ever decided to further develop the application into future sprints, we would start here.

The furthest developed feature that was never implemented was most definitely the notification system. It was being worked on from the very beginning of development and was constantly being tested. The notification system included automatically sending emails regarding to confirmation of ticket purchases, reminders, upcoming events, recent followers and much more. The code was completed however never integrated, because of how close we were to completion we would begin here with no hesitation.

My Favourites was a feature that never made it into the development phase. It was a more refined version of My Feed, instead of including events from followed societies it would show events that were favourited. The original plan was on event details, alongside the “Join” button there would be a “favourite” or an icon of a heart next to it. This feature would be used when you want to bookmark and event that you may want to attend but you're not 100% certain yet.

A limit to how many students was going to be added as a component for event creation. Upon creating an event, you could set he maximum number of tickets that could be sold. This feature would be useful if the event took place in a studio which had limited seats or standing space to ensure the event was not overbooked. It was never implemented due to there not being backend functionality to keep track of how many tickets were sold.

Relating back to what I was mentioning in “3.5 Story Maps”, there were many ideas that were completely scrapped altogether. These ideas were scrapped because they weren't fundamental and too complex given the close deadline. These could possibly be implemented after the three developments I previously mentioned. This could include a RSVP system for confirmation of attendance, an event feedback forum and a voting system which allows students to interact and present new ideas to societies.

In conclusion, there are an endless number of features that could be further implemented into the application, we would first start with the features that we didn't get enough time to fully develop due to constraints and then features that were scrapped for being too complex. At that point, our application would be established enough to publish and receive feedback for additional features.

3.8 Code Versioning

For code versioning we all used the Gitlab tool for our code repository and collaborative code development. We found it had an easy and smooth implementation with our chosen IDE (Visual Studio Code). We could make push and pull requests directly from VS code and could easily sync it to our branch after we did so.

We decided that the easiest way to ensure proper implementation of our code and to avoid clashes was to follow our own branching strategy where each of us had our own branch to work on and update our developed features.

We also had a master branch that we called "main". Main would always have the most updated version of the code and in order to avoid merge crashed we followed the following rules as a team:

1. Before merging to main, make a pull request from source main to target (your branch) so you ensure you have an updated version of the code.
2. Any merge errors will be dealt with locally on your branch to avoid issues with the main branch.
3. When merging, always make sure to add a commit comment explaining what features you have altered and add a description if necessary.
4. When merging to main, assign the merge request to another teammate so at least someone else can code review that the merge won't cause any errors and that any errors manually fixed are properly implemented and embedded in the code.
5. When completing a merge request, announce it on Teams so all team members know there has been updates made to the code.

This allowed for parallel development in which we always had a bug-free fully functioning version of the code that could be pulled from main. Branches were only merged into main when a task had been fully completed by a team member and had been tested on their own branch to ensure proper functionality.

Tasks could be found in our Jira Board, where we maintained a backlog of tasks to be completed and added them to the sprint accordingly. The tasks were all liked on Jira to its relevant user stories, so the developers could check if their code's functionality worked accordingly to the team's idea prospect of it.

As we reflect on the way we handled branch management and commit mistakes, we acknowledge maybe another way we could've more easily merged our code to main was by using a branching strategy that was more feature focused, instead of relying exclusively on commit comments to resolve any issues that may have come up. We also made the mistake a few times of deleting the source branch after a commit to main. Going forward, these are things to be more mindful of.

3.9 Team communication/management

The way we structured our team consisted, mostly, of front end and back-end development. Initially each of us voiced what part of the project we would like to work on. After some compromise we agreed that the sub-teams were as follows:

<u>Backend</u>

- Patrick Johnson Aggrey
- Sam Olatilewa Ishola
- Chikezie Ogbogu

<u>Frontend</u>

- Nathan Kenneth O’Connor
- Dmitrijs Kraliks
- Anshuk Reddy Gaddam
- Paula Catalan Santana

As a team we appointed Dmitrijs Kraliks as our Product Owner given, he was the one who’s proposal we ended up using and had the clearest idea of the scope of the project. His role mainly consisted in resolving any conflicting ideas or proposals for features and to guide us in the right direction as we were developing the code.

As the weeks went on and either sub-team required more attention at different stages of the development, some members from frontend aided their backend peers and vice-versa, we worked in a very flexible way when it came to our roles.

Our communication was mainly through our “team11” channel on Microsoft Teams. Here we shared all our documents and communicated on updates regarding the project. Our meeting minutes were shared as a shared Microsoft Word document and each week the minutes taker would update it as meetings took place. Any concerns that needed a prompt resolution were also shared on the team’s channel as well as updates and suggestions. Sometimes, we would use the channel to create polls in order to decide things such as when meetings would take place.

Another communication tool we heavily used was our Jira board. Here we could update our tasks and subtask’s staging and progress. Tasks would be moving through stages such as 'To-Do,' 'In Progress,' 'Code Review,' 'Testing,' and finally, 'Done’, allowing for a very visual representation of the project progression. Jira allowed us to assign tasks to different team members and to link the tasks to user stories. Jira also offers a report feature where we able to download all the report graphs shown in previous sections that have allowed us to have a more informed history of our scrum progression.

We had, for the most part, biweekly meetings, this included Wednesdays at 11 o’clock during our lecture slot for CS353 and Fridays online at 10 o’clock. As we used Agile SCRUM project management framework, we had a daily stand up during both our meetings where we would cover our progress shown on the Jira board and any roadblocks that might prevent us from moving forward. Every week we chose a new scrum master and minutes taker. Everyone in the team held both roles at least once.

We worked in two-week sprints, at the start of which we would have a Sprint Planning session. We would sit as a team and look through our backlog and decided what was needed to be done during the next sprint and in what priority order. We also assigned the tasks accordingly. This is when we would set the sprint goal.

Once we completed the sprint, we would have a Sprint Review meeting where we would show each other our progress within the app and explain our accomplishments of the last two weeks. Once this meeting was complete, we would engage in a Sprint Retrospective, in which we reviewed what went well (or what didn’t work) during the sprint and how to improve moving forward.

3.10 Remote working

Our remote work was mostly focused on code development and our Friday online meetings at 10:00 AM. If for some (valid) reason someone couldn’t attend the online meeting or if the meeting needed to be rescheduled, it would all be posted on teams. Any issues or updates would also be posted on the team’s channel. As a team, we all agreed to try our very best about interacting with the posted messages, especially to assure messages had been communicated properly and had reached all team members.

Our Friday meetings also took place on the Microsoft Teams platform, using their online videocall feature. Generally, the scrum master for the week would oversee the setting up of the meeting. He or she would also take initiative to send a reminder message the day before a meeting was to take place so everyone would attend. Often, a feedback message was usually sent after every meeting to express good work and effort being put into the project. This was some positive reinforcement that allowed for the group moral to be positive and productive throughout the project. This constant feedback created a sense of appreciation among team members and motivated everyone to stay engaged and contribute consistently.

Our Jira board was essential in making remote work successful. Some common questions that arose during the development of our code included *who is working on a *x* feature* and *what should I work on next*. Instead of broadcasting a message to the entire team and waiting for a reply from everyone, team members could find these answers on the Jira board. All tasks were assigned to at least one team member and the list of tasks left in the backlog, with task priority, were accessible by everyone in the team.

Google docs also was a big part of our remote work as it was used to share the meeting minutes and later, the collaborative parts of the report. The platform allowed us to edit and update meeting minutes during our virtual meetings which allowed for everyone to be able to access them and ensure all the relevant information was being added. It made so that everyone could at any given time, access what had been discussed during every meeting. It became a good helping tool for those who missed a meeting and needed to catch up.

Remote working allowed for a lot of flexibility which was advantageous as all team members have different schedules. Some challenged it posed, however, was some communication issues at times. It is hard to keep 7 people actively updating on their progress and replying to roadblocks.

However, by sticking to our biweekly meetings, making sure we added comments on every Jira board update, and fostered a good working environment with motivational messages and positive reinforcement we mitigated most of our communication issues.

By the end of the project, we became more agile and proactive when it came to addressing most of our communication challenges. As the project moved forward, we steadily improved our coordination and strengthened our ability to collaborate successfully as a remote team.

Chapter 4: My Contribution to the Project

Initially my contribution to the project was as a front-end developer, however, even though I started by only focusing on my assigned tasks, I eventually started working on some back-end development so as a team we could meet our deadlines.

My first task was to develop the application’s ‘Home’ page. The initial implementation of the ‘Home’ page included a title, search bar, event type filtering button and an event grid that would show all events offered on the application. As the back-end team was working on other areas of the project, for visualisation purposes I used manual data to fill in the grid and test its functionality. The grid consists of a container that contains individual event cards, all clickables. One of the new skills I learnt, was how to use the map () method where data is set one element at a time; I used an article recommended by ***ChatGPT (Bibliography 4.1)*** to learn how to implement this: “***Filtering data in React: `filter()`, `map()`, and `for` loops” (Bibliography 4.2)***. One of the biggest challenges was the implementation of the search bar and type event, visually building them wasn't so much the problem, it was creating a constant response as the user updated the search bar or clicked an alternative option for the button. I found two articles, ***"Search Bar in React JS!" (Bibliography 4.3)*** and ***"Build a Search Component in React" (Bibliography 4.4)*** that were really useful regarding the search bar implementation, interactivity and reactivity.

My next assignment was the front-end development of the 'MyFeed' page. This page worked and looked very similarly to the home page, so the only new development needed was to change the backend functionality of the grid itself as instead of showing all the events offered in the application, it should only show those events that are hosted by the users followed societies or students. For the most part the CSS used was searched for in the ***"w3schools CSS Tutorial" (Bibliography 4.5)*** or if something was completely outside my knowledge regarding its existence, I would often refer to ***ChatGPT (Bibliography 4.1)*** for suggestions on how to resolve given issues (i.e. How to not allow text to overflow outside of a container).

At this point of the development, we realised we needed to redesign the 'Navigation Bar' as it would appear on all the application pages, so I got to work on that. I added the image space for the app logo and all the buttons that would reroute the user to their designated pages. I also included a log out button so the user could log out of their account. As I was completing this, we also started integrating the different user types and account types into the structure and architecture of the project. This meant that depending on the account type, we would have to hide or show certain parts of the website and pages. This was challenging as I wasn't quite sure how to approach this issue. After researching, I came across the use of handlers (***"How to show/hide Modal in React" (Bibliography 4.6)***) which allowed to me show different parts of a component depending on the page that the user was on and depending on the account type.

Particularly for the 'Navigation Bar' component, if the user account was 'Student' all the buttons would be shown i.e. Home, MyFeed, MyTickets, Societies and LogOut. However, if that account type was 'Society' or 'Union', the buttons on the 'Navigation Bar' would just be Home, MyProfile, Societies and LogOut.

The next task I worked on was the Societies page, known in our repository as 'List Societies'. For this page I created a new component called 'SocietiesGrid' and reused my search bar component mentioned before. The SocietiesGrid mirrors the Home page 'EventGridDiscover' component as it uses the same styling. However, the fields were edited on each event card (which in this page was renames as society card) so they would only show the society name, email and short description. For the first time in the project, I worked on the backend of a page. Having never used react nor firebase before this project, I found the ***"Firebase in React: Step-by-Step Guide" (Bibliography 4.7)*** extremely useful as it had a step-by-step description on how the firebase structure worked for collections. It also contained suggestions on how to request and push data from and to the collections. Finally, one of my teammates edited the rerouting of the clickable society cards, so once clicked it would take the user to that society's profile. This was done by passing on the *societyId*.

As I became more comfortable working with backend, I began helping the backend team as most front-end components had been fully developed by then. I was given the front-end for 'EventDetails' page that acts like a popup. I fixed some css bugs that I came across but mostly worked on how the two buttons at the bottom of the popup "Back" and "Buy Ticket" worked. For the "Back" button it was a simple case of using Firebase's useState built in functionality, if the popup was set to open it would show (when the event card was clicked the state of popup became Open). When the "Back" button is clicked, the state of the popup becomes Close, so the pop up is closed. When the popup is first triggered by the clicking of an event card, Event Details is called. The parameters passed into event details are 'eventId', 'onClose' and 'isMyTicketsPage'.

The last parameter was added later on so I will explain further down in the report. The event id is the id given to the event clicked, this is used to open the popup 'Event Details' which will then show more information about that event including, a longer description, the date and the location. Parameter 'onClose' is the callback function 'handleClosePopup'. It allows the 'Event Details' component to notify the parent to close the popup when clicking the "Back" button.

The 'Buy-Ticket' button communicates heavily with the back end (firebase NoSQL database). When the said button is clicked, the event id is sent to the "joinedEvents" collection and added at the end of the array eventsJoined where the email matches the logged in user's email. In the case where the user is joining their first event, a new element is created in the collection with the user's email and the id of the event as the first event id on the eventsJoined array. When this transaction takes place, the Buy-ticket button displayes Joined, becomes grey and is disabled.

During this time, the Home page and MyFeed page's back end had been set up by one of my peers. That meant I could now integrate Event Details into both pages. As explained before, every event card is populated as information on the event is pulled from the events collection from our firebase database. When the event is clicked it triggers the Event Details popup.

Once this was complete, my next task was to implement backend of the created MyTickets page. When we first merged the file into the project, we encountered some CSS issues as this page's front-end was developed differently to the rest. Links were used initially to use some online posted CSS; this was affecting the entire application's styling so I first redid the CSS so it would look like the rest of the app and work appropriately. Once that was done, I moved on to the back-end component of the page. When the code was implemented the paged worked as follows: the logged in user's email is used to fetch from the 'joinedEvents' collection the array of id's of events joined. Each id is then searched for in the 'events' collection and the information is displayed as a ticket in a column like way on the MyTickets page. Each ticket is clickable. I quickly realised that when the Event Details popup opens the "Buy-ticket button" which now displays "Joined" as it is a joined event, still appears unnecessarily. That's when I added the extra parameter that goes into the component EventDetails when it is called, 'isMyTicketsPage'. If and only if, the component is triggered from the MyTickets page then the "Buy-ticket button" no longer appears, and the Back button is centred.

My second to last task was to make the "Live events" events on MyProfile, clickable and so they triggered the Event Details popup. Once again, the implementation was extremely similar to the last few tasks I had worked on, but I had to add the events into individual containers and create responsive CSS for when they were hovered over and clicked on.

My last task consisted of implementing the developed "Start Page" so that when the app was ran, it was the default page it would take the user to. When I first integrated the start page, the carousel that switched through 5 images on the background wouldn't recognise the array of images being called from a different folder. Eventually I used a different way of calling the images, by calling them individually and then adding them to an array and it seemed to do the trick as the carousel was there after working properly. I found this method in ***"Stack Overflow" (Bibliography 4.8)***.

Chapter 5 Summary

Working in a team

Working in a team is something we are used to as university students, but I believe this experience has been the closest to a real-life work environment. It is always hard to keep seven people on the same page and working to their full potential for 12 weeks, but I think as a team we did quite well. Some weeks we would find some team members begun to disappear in the crowed a bit, but I believe the positive communication and encouragement during meetings helped us to maintain a good attitude, attendance rate and rate of development.

Most of us, if not everyone, seemed really excited about the product as we could visualise it in a real-world setting. Not only that, but we all had first-hand experience with a similar platform, and we knew what features didn't work for us which led us to have a very clear idea of how we wanted this application to work.

I really enjoyed participating in this project as I think it has taught me patience within a team and compromising. I am proud of the product we have presented, and I believe we followed an agile way of development that allowed us to adapt our application as we went deeper into the development stages and found theoretical ideas that didn't work in practice or found better and more efficient alternatives.

Things that went well

Some of our strongest attributes were our participation and communication. In some capacity, we all added participated in the project and our input was essential for the project to become what it ended up being. For the most part, we had a full house during every meeting and people generally participated and took seriously their assigned roles for the week (i.e. Scrum master and/or minutes taker).

Our communication was often and efficient. Any queries were always answered within around 48h and everyone felt comfortable sharing new ideas or concerns.

Development was also quite successful. Code was continuously developed, tested and deployed. No one was afraid of learning a new skill, even though most of us were not familiar with react, firebase or even GitLab, we all took upon ourselves to learn independently and complete tasks.

Another positive skill we developed and enforced was error handling. Whenever there was an evident bug, more than one developer would take a chance at debugging it. And if at any point any team member encountered an error, glitch or malfunction of any of the components or pages (even if not their own developed work) they would raise the concern and another team member would usually offer to fix it.

Things that went wrong

On the other hand, a few things could've been improved in hindsight. Workload distribution was very unrealistic the first two or three weeks in my opinion. Our estimation development wasn't always accurate. I am proud to say that as the weeks went by and we had a better feel of our skills and time management we got better and better at estimating completion time and assigning tasks in the different sprints.

I believe a little more research into GitLab at the beginning of the project would've allowed for a more efficient development process. Using established branch strategies such as "GitFlow" would've probably aided in our branch merging, as having a more component based merging strategy would've probably allowed for easier spotting of errors and ease of implementation.

In order to avoid spending too much time in the designing stage and not starting the development of code early enough, I believe we made the mistake of under designing our components and backend structure. Front end architecture and backend architecture didn't always match at the start which forced some redevelopment further down the line. Although some redesigning is useful and allows for agile workflow, I believe the amount of code and collection communication we had to redesign was excessive and probably hindered our progress for a couple of weeks.

Use of AI tool for research and debugging

AI was a key feature during this project for me. Learning how to prompt engineer to get the best results was probable the most useful skill I have learnt so far regarding AI. I wanted this project to be a deep learning experience for me so avoiding the copy/paste of code from an AI chatbot was essential. Instead, I strived to make my use of AI strictly informative and for error spotting when I decided I had spent enough time trying to debug the code myself.

Having never used react nor firebase before, using an AI chatbot allowed me to quickly learn a lot of information. In the case I didn't understand at first, it was very easy to ask for further or an alternative explanation which made the learning experience very comfortable.

The use of examples was also key, as seeing built in methods and their syntax in real code examples helped me understand their structure and interaction with other elements of the code. If some of my peer's commented code wasn't clear enough, this was also a great tool to break down the code and build on top of it.

I do recognise AI is error prone, especially when it comes to complex code explanations, but understanding the current limits of AI and contrasting information with other resources avoided any major confusion or misunderstanding.

All in all, AI was an amazing tool to work with, but I tried my best to use it appropriately and efficiently.

Conclusion

In conclusion, I can confidently say this project (and per extension module) has been one of the most hands on and informative from my degree so far. I learnt how to communicate within a small group, learnt compromise and to value how other people might depend on my work. This last lesson made me reflect on how I should take pride in delivering the best version of my work possible, not just for me but for the good of the product and the team.

I think the only change I would make regarding the team project as a whole is probably the amount of team members. I think 5 team members would've sufficed and the more people within a team the harder it is to communicate and manage. I would like to make a point that demonstrators were essential throughout the project and resolved many queries of ours.

Overall, a great experience. It has been challenging at times but worth it as we have built a product, we are very proud of.

Appendix:

Meeting minutes:

Team 11 meeting - 02nd Oct 2024

Name	Attendance
Anshuk Gaddam	Y
Nathan	N

Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

Initial ideas:
<u>Anshuk Gaddam *FrontEnd*</u>
1. Sleeping app, tracking sleep and suggesting a sleep pattern

<u>Patrick Johnson Aggrey *BACKEND*</u>
N/A
<u>Chikezie Ogbogu *UI*</u>
N/A
<u>Sam Olatilewa Ishola (*BACKEND*)</u>
SCRUMMASTER WEEK 1
1. Mindfulness or motivational app to reach your goals. Network with people who are also struggling or dealing with same event. Suggestion for events related to content or goals you want to pursue. “Goaler”

2. “Society” Tinder for finding friends that have similar interests to you.

<u>Paula Catalan (*FrontEnd*)</u>
1. Planner app; to do lists, event calendar, organisational app

<u>Nathan</u>
1. Maynooth marketplace, either buy/sell what you need as a student or create a small business to sell stuff around college.

<u>Dmitrij (*Frontend*)</u>
Product Owner
1. Event booking app for su/ Maynooth events
2. SEO ranking tool (web scrapper) Google analytics etc Give a SEO score
3. “Fake indeed”
4. Stripe API websites

Main idea chosen:
Name: Uni Life
Description: Maynooth Events App for societies, SU and student lead events.
Technology:
- Backend: Firebase / MongoDB
- FrontEnd: Javascript/React (Typescript) / Bootstrap

Bring to next meeting:
Item 1: Write a breif 1 paragraph summary on what the system is.
Item 2: Write user scenarios for all user types of the system.
Item 3: Construct the backlog of index cards (user stories)
Deliverables: These must be uploaded to Moodle. You can take a photo/screendump of the index cards and include those in the upload.
Meeting started: 11:00 Am
Meeting ended: 13:35 Am

Team 11 meeting - 04th Oct 2024

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan	N
Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

*Nathan was absent due to work commitment
Minutes keeper:
Paula Catalan
Points mentioned:

- First meeting set up.
- For next week complete user stories and paragraph on understanding of project.
- Individually add requirements you believe should be part of the project, bring to Wednesday 9th October meeting.
- Wednesday 9th we will plan our first sprint.
- Sprint review will happen every 2 Wednesday during the meeting (At the start of the meeting)
- Sprint Review is a formal meeting between scrum teams that happens at the end of a sprint. In sprint review meetings, the team meets to show their work and review it together.
- First sprint review: Wednesday 23rd of October
- Sprint retrospective every two Fridays at the start of the meeting.
- The Sprint Retrospective meeting happens after the Sprint Review meeting. Sprint Retrospective is the last meeting of the sprint or cycle.
- First sprint retrospective: Friday 25th of October
- Watch Jira tutorial for next Wednesday, posted on teams's channel.

Meeting started: 10:00 Am
Meeting ended: 10:40 Am

Team 11 meeting - 09th Oct 2024

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y

Nathan	Y
Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	N
Paula Catalan	Y

*Patrick missed his train

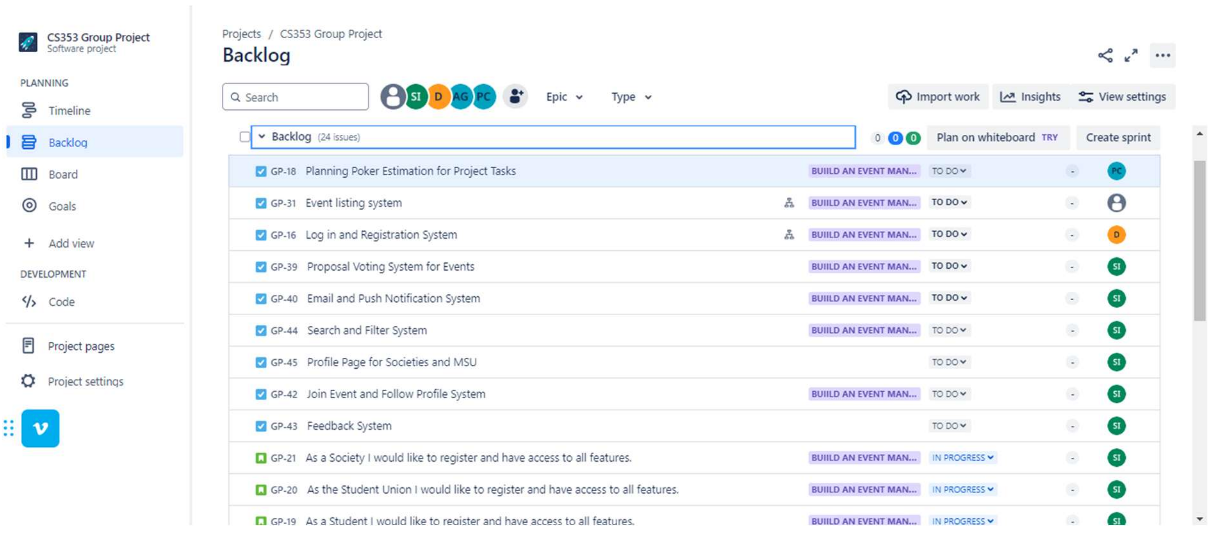
Points mentioned:

1. We started the meeting by gathering our user stories and compiling them into one general file.
2. Student stories were divided into categories
3. Then we added them as requirements to the backlog in Jira. Anshuk added everyone's user stories into one big file and filtered through, leaving us with non-repeating stories. Then the stories were sorted into "sections" or categories.
4. Meanwhile, Chikezie kept working on optimizing the initial homepage design so we could decide which tasks to tackle.
5. Paula, Sam and Dmitrij worked on adding tasks to the backlog as the user stories were being sorted.
6. Nathan joined Jira.
7. We discussed the best way to divide the tasks so we could use planning poker to estimate the working time for each.
8. The whole team then worked on adding tasks to the backlog on Jira.
9. Process for user:

User registers and logins in.

After registering the user is prompted by a survey to which they select societies of interest which are later presented on the users home page (like a Instagram feed) alongside the msu events. The msu will have its own navbar position since its unique which will have its events There will be a search option so that the user can search and filter events with tags and filters and societies. The societies will have their own pages that can be accessed and you can see specific events from that society. User can browse their tickets and leave reviews on events that they have registered for previously that have finished.

Jira Backlog



Backlog

Backlog (24 issues)

GP-18	Planning Poker Estimation for Project Tasks	BUIILD AN EVENT MAN...	TO DO	PC
GP-31	Event listing system	BUIILD AN EVENT MAN...	TO DO	
GP-16	Log in and Registration System	BUIILD AN EVENT MAN...	TO DO	D
GP-39	Proposal Voting System for Events	BUIILD AN EVENT MAN...	TO DO	SI
GP-40	Email and Push Notification System	BUIILD AN EVENT MAN...	TO DO	SI
GP-44	Search and Filter System	BUIILD AN EVENT MAN...	TO DO	SI
GP-45	Profile Page for Societies and MSU		TO DO	SI
GP-42	Join Event and Follow Profile System	BUIILD AN EVENT MAN...	TO DO	SI
GP-43	Feedback System		TO DO	SI
GP-21	As a Society I would like to register and have access to all features.	BUIILD AN EVENT MAN...	IN PROGRESS	SI
GP-20	As the Student Union I would like to register and have access to all features.	BUIILD AN EVENT MAN...	IN PROGRESS	SI
GP-19	As a Student I would like to register and have access to all features.	BUIILD AN EVENT MAN...	IN PROGRESS	SI

Bring to next meeting:
Meeting started: 11:00 Am
Meeting ended: 1:40 Pm

Team 11 meeting - 11th Oct 2024 (Online)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan	Y
Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	N

Paula was at an interview*

Points mentioned:

- Discussing what has been started on
- Plan to do story points and spilt tasks
- Firebase was decided on for back end before meeting in a poll
- React is being used for front end
- Planning for what to do in this current sprint
- Having a functional login and registration system assigned to Dmitrij
- Properly assigning back end and front end
- Discussing how code will be shared
- Gitlab will be set up on Wednesday
- Deciding on Preset Societies

- User registration survey front end assigned to Anshuk
- Assigned tasks to everyone
- Discussing more on how the user registration survey works (it should on be asked to students)
- User registration survey back end assigned to Chikezie
- Discussing how people will login, Student number etc for students, Email for societies
- Estimated how long each task would take
- Will discuss in more detail when Paula is back

Jira Backlog
N/A
Scrum master: Sam Olatilewa Ishola
Minutes keeper: Chikezie Ogbogu
Bring to next meeting:
Gitlab setup

Meeting started: 10.00 Am
Meeting ended: 10.55am

Team 11 meeting - 16th Oct 2024 (In-Person)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan	Y
Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

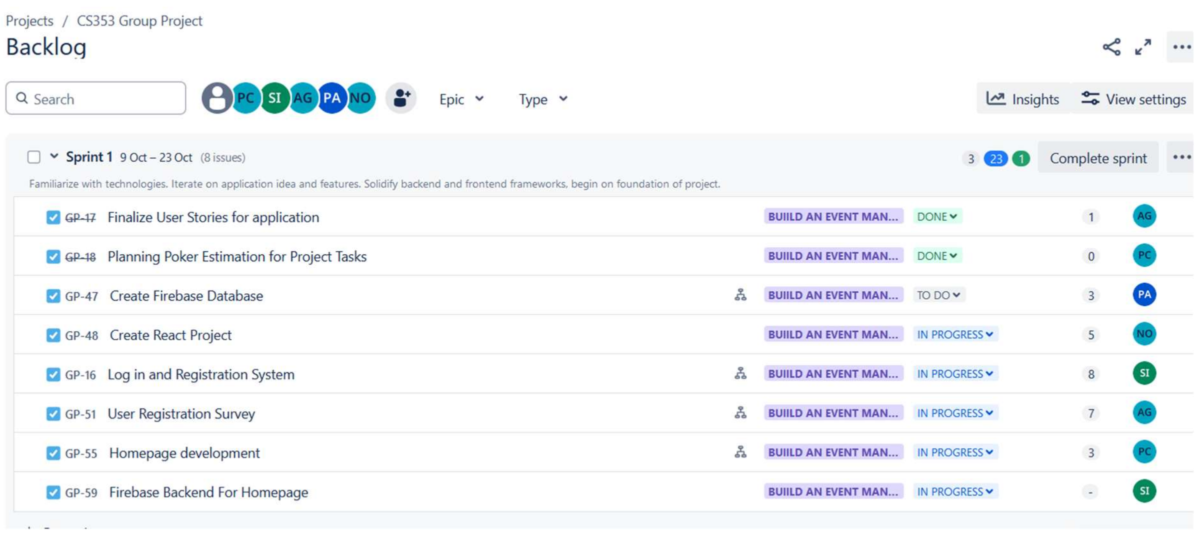
Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	Wireframe for student registration form	Working on the student registration form	
Nathan	Wireframe for student registration form	Working on the student registration form	
Dmitrij	Login and registration System Core	Testing and debugging making sure login and	none

		registration system works as expected	
Sam Olatilewa Ishola	created Gitlab Repository for collaborative development and created Jira project for project management		
Chikezie Ogbogu	Learning Firebase		none
Patrick Johnson Aggrey	Story map		
Paula Catalan	Researched React's framework	Researched React's framework and built the initial setup for the "Home" page.	Issues: suggested new technology but as a group we decided against it Moving forward with OG technology.

Points mentioned:

- Gitlab was set up prior to this meeting by Sam, Sam added anyone who wasn't on it already, on it
- Nathan and Anshuk worked on a wireframe and will be implementing it before the end of the sprint
- Paula worked on installing react and so far, was gotten a navigation bar working
- It was decided that Sam will do the backend for the home page
- Dmitrij worked on login and registration system and is not testing its functionality
- Patrick worked on a story map
- Backend Developers were added to the firebase
- As team we set up Gitlab and its branches, we tested some test commits and merge requests to ensure that it was working properly. We came across some issues when cloning the branches but fixed it once we understood we could switch from main to a different branch directly from VS code.

Jira Backlog



Projects / CS353 Group Project

Backlog

Search · PC SI AG PA NO · Epic · Type · Insights · View settings

Sprint 1 9 Oct – 23 Oct (8 issues) 3 23 1 Complete sprint

Familiarize with technologies. Iterate on application idea and features. Solidify backend and frontend frameworks, begin on foundation of project.

~~GP-17~~	Finalize User Stories for application	BUIILD AN EVENT MAN...	DONE	1	AG
~~GP-18~~	Planning Poker Estimation for Project Tasks	BUIILD AN EVENT MAN...	DONE	0	PC
GP-47	Create Firebase Database	BUIILD AN EVENT MAN...	TO DO	3	PA
GP-48	Create React Project	BUIILD AN EVENT MAN...	IN PROGRESS	5	NO
GP-16	Log in and Registration System	BUIILD AN EVENT MAN...	IN PROGRESS	8	SI
GP-51	User Registration Survey	BUIILD AN EVENT MAN...	IN PROGRESS	7	AG
GP-55	Homepage development	BUIILD AN EVENT MAN...	IN PROGRESS	3	PC
GP-59	Firebase Backend For Homepage	BUIILD AN EVENT MAN...	IN PROGRESS	-	SI

Scrum master: Paula Catalan
Minutes keeper: Chikezie Ogbogu
Bring to next meeting:
Gitlab

Meeting started: 11.00 am
Meeting ended: 1.00 pm

Team 11 meeting - 19th Oct 2024 (Online)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan	Y
Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

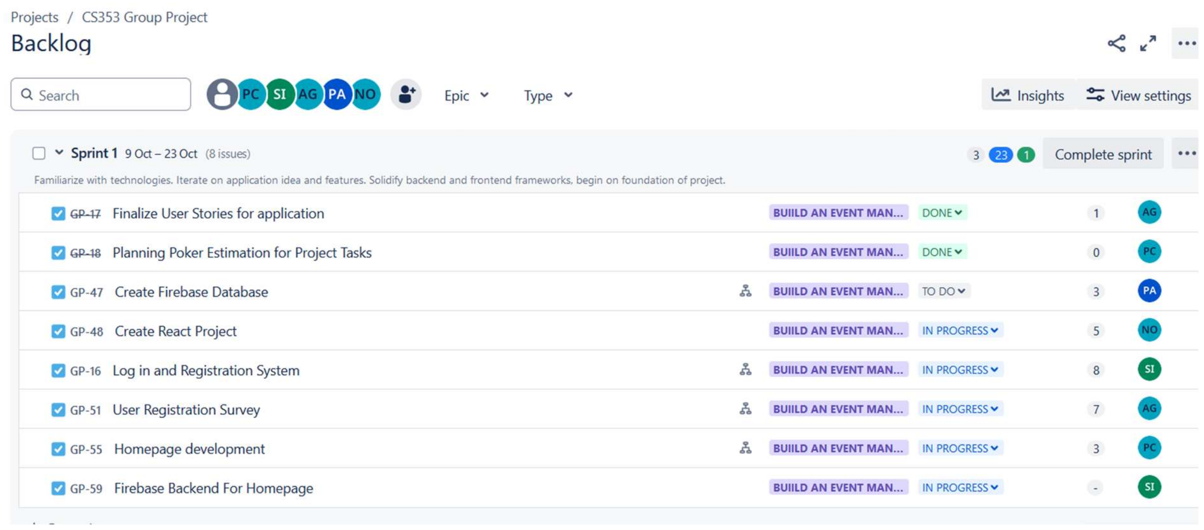
Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	Wireframe for student registration form	Working on the student registration form	
Nathan	Wireframe for student registration form	Working on the student registration form	

Dmitrij	Login and registration System Core	Testing and debugging making sure login and registration system works as expected	none
Sam Olatilewa Ishola	Set up Gitlab, created and cloned branches for each member		
Chikezie Ogbogu	Learning Firebase		none
Patrick Johnson Aggrey	Story map		
Paula Catalan	Homepage Framework	Complete the first menu on the homepage. Search bar and Toggles	

Points mentioned:

- Checked up on what everybody was working on
- Essentially the same things as the late meeting
- Discussed using Gitlab
- Scheduled a meeting for the backend side

Jira Backlog



Projects / CS353 Group Project

Backlog

Search | PC SI AG PA NO | Epic | Type | Insights | View settings

Sprint 1 9 Oct – 23 Oct (8 issues) 3 23 1 Complete sprint

Familiarize with technologies. Iterate on application idea and features. Solidify backend and frontend frameworks, begin on foundation of project.

~~GP-17~~	Finalize User Stories for application	BUIILD AN EVENT MAN...	DONE	1	AG
~~GP-18~~	Planning Poker Estimation for Project Tasks	BUIILD AN EVENT MAN...	DONE	0	PC
GP-47	Create Firebase Database	BUIILD AN EVENT MAN...	TO DO	3	PA
GP-48	Create React Project	BUIILD AN EVENT MAN...	IN PROGRESS	5	NO
GP-16	Log in and Registration System	BUIILD AN EVENT MAN...	IN PROGRESS	8	SI
GP-51	User Registration Survey	BUIILD AN EVENT MAN...	IN PROGRESS	7	AG
GP-55	Homepage development	BUIILD AN EVENT MAN...	IN PROGRESS	3	PC
GP-59	Firebase Backend For Homepage	BUIILD AN EVENT MAN...	IN PROGRESS	-	SI

Scrum master: Paula Catalan
Minutes keeper: Chikezie Ogbogu
Bring to next meeting:

Meeting started: 9.55 am
Meeting ended: 10.16 am

Team 11 Backend meeting - 20th Oct 2024 (Online)

Meeting attendance

Name	Attendance
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y

Name	Work Done	Currently Working on	Issues or blockage
Sam Olatilewa Ishola			Firebase credentials
Chikezie Ogbogu			none
Patrick Johnson Aggrey			

Points mentioned:

- Discussing what was been work on so far
- Assigning specific tasks isn't possible at this point due to the app not properly working
- Making sure everyone can open the workspace
- Going to focus on working together until we have solid foundation
- Files are so app isn't working properly
- Focus on getting the app working
- Main goal get apps working on local machines
- Installing dependencies for backend
- Planning for iterative development

Scrum master: Sam Olatilewa Ishola
Minutes keeper: Chikezie Ogbogu

Bring to next meeting:
Meeting started: 8:45 pm
Meeting ended: 9.26pm

Team 11 meeting - 23rd Oct 2024 (In-Person)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	Y

Dmitrijs Kraliks	N *
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

*Dmitrij's at an interview

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	Wireframe for the student registration form	Working on the student registration form	
Nathan			
Dmitrij	Login System working and functioning correctly, password hashing	Working on user profiles page	none
Sam Olatilewa Ishola	Users and societies collection creation in firestore		none
Chikezie Ogbogu			none
Patrick Johnson Aggrey			
Paula Catalan	Complete the first menu on the homepage. Search bar	Home page, working on toggles and grid for events	none

Points mentioned:

- Discussed the roadblocks in backend with the rest of the group
- Had to remake branches as the main branch was changed
- Paula finished the search bar before this meeting
- Going through Jira tasks
- Checking which ones are done or still in progress
- First Sprint completed!!!
- Merged Paula branch into the main branch
- Created Anshuk's user so he can upload his commits to the branch
- Nathan and Anshuk have updated the attributes of the components
- Started Sprint 2
- Assigned future tasks for when current tasks are finished
- Planned for splitting backend tasks in the future
- Voted for Scrum Master and Minutes Keeper

Scrum master: Paula Catalan
Minutes keeper: Chikezie Ogbogu
Bring to next meeting:

- On Friday:
- VM setup decision
- Who and how to change the physical user storymap, ITS GRADED (Jira not enough)

Meeting started: 11:00 Am
Meeting ended: 1:15 Pm

Team 11 meeting - 25th Oct 2024 (Online)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	Y
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

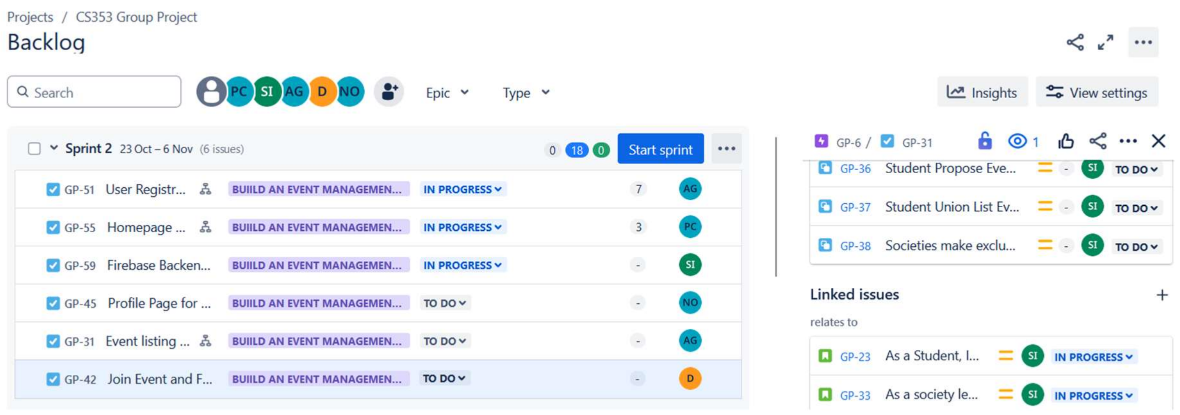
Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam		Event Listing System	none
Nathan			
Dmitrij		Working on user profiles	none
Sam Olatilewa Ishola		Database expansion	none
Chikezie Ogbogu			none
Patrick Johnson Aggrey			
Paula Catalan	No major updates	Home page, working on toggles and grid for events	none

Points mentioned:

- Discussed what each team member is currently working on and any alerts for bugs or blockages. (Daily Stand-Up)
- Vote for reading week meeting
- Next Backend meeting
- User Story map needs to be regularly updated
- Virtual machine will not be set up until backend running smoothly.

Scrum master: Anshuk Gaddam
Minutes keeper: Sam Olatilewa Ishola
Bring to next meeting:

Meeting started: 9:55 Am
Meeting ended: 10:15 Am


Projects / CS353 Group Project
Backlog

Sprint 2 23 Oct – 6 Nov (6 issues) 0 18 0 Start sprint

GP-51	User Registr...	BUIILD AN EVENT MANAGEMEN...	IN PROGRESS	7	AG
GP-55	Homepage ...	BUIILD AN EVENT MANAGEMEN...	IN PROGRESS	3	PC
GP-59	Firebase Backen...	BUIILD AN EVENT MANAGEMEN...	IN PROGRESS	-	SI
GP-45	Profile Page for ...	BUIILD AN EVENT MANAGEMEN...	TO DO	-	NO
GP-31	Event listing ...	BUIILD AN EVENT MANAGEMEN...	TO DO	-	AG
GP-42	Join Event and F...	BUIILD AN EVENT MANAGEMEN...	TO DO	-	D

GP-6 / GP-31

GP-36	Student Propose Eve...	SI	TO DO
GP-37	Student Union List Ev...	SI	TO DO
GP-38	Societies make exclu...	SI	TO DO

Linked issues

relates to

GP-23	As a Student, I...	SI	IN PROGRESS
GP-33	As a society le...	SI	IN PROGRESS

<u>Team 11 Backend meeting - 27th Oct 2024 (Online)</u>

Meeting attendance

Name	Attendance
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y

Name	Work Done	Currently Working on	Issues or blockage
Sam Olatilewa Ishola		Societies collection and subscribedProfiles field for Student Users	none

Chikezie Ogbogu		Notification system with FCM	none
Patrick Johnson Aggrey		Event creation CRUD	none

Points mentioned:

- Delegating work
- Everyone has access to firebase
- Sam working on Society collection and CRUD operations for members of societies
- Task for event creation and CRUD functionality delegated to Patrick
- Task for Notification System with Firebase cloud messaging delegated to Chike
- Starting to code

Scrum master: Sam Olatilewa Ishola
Minutes keeper: Sam Olatilewa Ishola

Discussed today:
Backend smoothness of operation
Making sure everything is linked correctly
Area of work tasks delegated to team members
Backend work trajectory

Bring to next meeting:
Meeting started: 9:00 pm
Meeting ended: 10:00pm

Team 11 - 30th Oct 2024 (Online)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	N
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	N
Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	Front End for User Survey	Event Creation pages for 3 main user types	none
Nathan			

Dmitrij	Started developing user pages	Need to adjust login system for societies to populate the society collection on registration	none
Sam Olatilewa Ishola	Code for populating soc collection	Societies collection and survey function population subbed socs for student users	none
Chikezie Ogbogu			none
Patrick Johnson Aggrey	Began working on the event creation backend	Code for population of the events collection	none
Paula Catalan	Home page, working on toggles and grid for events	Finish implementing the toggles for the final menu. Add link from search bar to grid?	none

Scrum master: Anshuk Gaddam
Minutes keeper: Sam Olatilewa Ishola
Discussed today:

- Populating the societies collection
- Homepage features
- Firebase Cloud Messaging for Notifications
- Profile Pages
- Event CRUD operations
- New Scrum Master and Minute Keeper

Bring to next meeting:
Meeting started: 11:00 Am
Meeting ended: 12:00PM

Team 11 - 06th Nov 2024 (In-person)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O’Connor	Y
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y

Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	Front End for User Survey	Event Creation pages for 3 main user types	none
Nathan			
Dmitrij		Working on user pages/profiles	none
Sam Olatilewa Ishola	Routing from account creation processes changes made to login and registration to integrate and route to surveys and so on	Backend integration for starting processes. Implementing firebase authentication for user management.	none
Chikezie Ogbogu		Setting up notifications	none
Patrick Johnson Aggrey		CRUD functionality for backend	none
Paula Catalan	Grid implementation complete.	Add filtering to grid	none

Scrum master: Dmitrijs Kraliks

Minutes keeper: Anshuk Gaddam
Discussed today:

- Paula added changes to the home page
- Dimitri done the sprint review
- Went over everyone's tasks in the sprint and reviewed each other's work
- Checking which ones are done or still in progress
- Second Sprint completed!!!
- Discussed how to populate the back end for the 3 main user types.
- Started Sprint 3
- Assigned future tasks for when current tasks are finished
- Patrick is going to update the user story map every week

Bring to next meeting:
Meeting started: 11:00 Am
Meeting ended: 1:00PM

Team 11 - 13th Nov 2024 (In-person)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	Y
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	Event Creation pages for 3 main user types and the user registration form.	Ticket page for events	none
Nathan			
Dmitrij		Working on user profiles	none
Sam Olatilewa Ishola	Route for survey submission	Routing from registration pages to survey pages and user authentication for staying logged in after registration	none
Chikezie Ogbogu			none
Patrick Johnson Aggrey			none
Paula Catalan	Societies page close to completion. Switch button from "Discover" to "My Feed" working. My feed page is structured but needs interactivity changes.. Header Menu restructured (mostly CSS) to account for	Moving on to redesign interaction of Switch	none

	new "Switch" Button. Spent some time commenting the code so the backend team finds it easier to navigate.		

Points mentioned:
Scrum master: Dmitrijs Kraliks
Minutes keeper: Anshuk Gaddam

Discussed today:

- Paula has nearly finished the myfeed page, just 1 more error to be fixed
- Went over everyone's tasks in the sprint and reviewed each other's work
- Checking which ones are done or still in progress
- Discussed how to populate the back end for the 3 main user types.
- Nathan is going to update the user story map every week instead of Patrick from now on
- Anshuk finished the Event Creation System for the 3 main user types and changed the CSS for the user registration survey.
- Discussed how the ticket grid for events would look like.
- Drew a wireframe for it

Bring to next meeting:
Meeting started: 11:00 Am
Meeting ended: 1:00PM

<u>Team 11 meeting - 15th Nov 2024 (Online)</u>

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan	N
Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam			

Nathan			
Dmitrij		Working on user profiles	
Sam Olatilewa Ishola		Survey Routing	
Chikezie Ogbogu			
Patrick Johnson Aggrey		Working on linking between events backend collection and the societies collection	
Paula Catalan	Complete redesign of header (navbar) and buttons on wirefames	Implementation of new changes	None

Scrum master: Chikezie Ogbogu
Minutes keeper: Patrick Johnson Aggrey
Discussed today:

- Discussed how we'd handle the work in the last few weeks and decided to create a roadmap for it at the next in-person meeting.
- Decided to showcase and combine our current work in the next in-person meeting in order to create a draft of what it would look like as of now to properly gauge how much more progress needed to be done.

Meeting started: 9:56 am
Meeting ended: 10:16 am

Team 11 Backend meeting - 19th Nov 2024 (Online)

Meeting attendance

Name	Attendance
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y

Name	Work Done	Currently Working on	Issues or blockage

Sam Olatilewa Ishola	Changes made to survey pages	Route testing using Postman	none
Chikezie Ogbogu			none
Patrick Johnson Aggrey			none

Points mentioned:
- Talked about the work weve all done so far
- Discussed how integration would work and made sure everyone had that in mind while individually working on their code

Scrum master: Chikezie Ogbogu
Minutes keeper: Patrick Johnson Aggrey

Meeting started: 21:06 Pm
Meeting ended: 21:30 Pm

Team 11 meeting - 20th Nov 2024 (In-Person)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan	Y
Dmitrij	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam			
Nathan			
Dmitrij		Working on user profiles	
Sam Olatilewa Ishola	End to end integration. dynamically fetching from the societies	User authentication using firebase and and	Waiting on updated home pages to be pushed to branch so

	collection to populate the survey. Survey submission updating users collection documents.	logic for a logged in user experience	that i may pull and integrate the front end with the backend
Chikezie Ogbogu			
Patrick Johnson Aggrey			
Paula Catalan	No new completed components.	Continuation on last weeks work.	None

Points mentioned:
Scrum master: Chikezie Ogbogu
Minutes keeper: Patrick Johnson Aggrey

Discussed today:
Discussed some changes Sam made to the survey page.
Discussed how the user experience would differ for the different kinds of users(student/societies/union) and what we would have to change
Paula decided to slightly change the structure of the home page

Meeting started: 11:00 Am
Meeting ended: 1:00PM

Team 11 meeting - 22nd Nov 2024 (Online)

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	Y
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam			
Nathan			
Dmitrij		MyProfile Page	
Sam Olatilewa Ishola	Event Creation front end and backend integration	Testing correct documents, fields and collections are populated after event creation	
Chikezie Ogbogu			
Patrick Johnson Aggrey			
Paula Catalan			

Points mentioned:
Scrum master: Patrick Johnson Aggrey
Minutes keeper: Nathan O'Connor

Discussed today:
Appointed new roles, Patrick took the place of Scrum Master and Nathan as the minutes keeper. Navigation bar has been updated with minor changes as well as the home pages. Society pages now include search bars. Discussed page routing. My tickets page is currently in development. Profile page is making great progress. Survey has been fully completed. Login integration has begun development. Database for societies has been created and dynamically populated. We finished the sprint and set goals for next sprint which include dynamically fetching data from home page, finish user interface and complete required merge requests in person.

Meeting started: 09:58 Am
Meeting ended: 10:28 Am

Team 11 meeting - 27th Nov 2024

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	N *
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y

Patrick Johnson Aggrey	Y
Paula Catalan	Y

*Out sick

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam		My Tickets Page	
Nathan		Start page and User story mapping	
Dmitrij		MyProfile page	
Sam Olatilewa Ishola		User Session handling and Application flow	
Chikezie Ogbogu		Notifications and Alerts	
Patrick Johnson Aggrey		MyEvents CRUD	
Paula Catalan	Alpha User Interface	Navigation routes on interface	None

Points mentioned:
Scrum master: Patrick Johnson Aggrey
Minutes keeper: Olatilewa Sam Ishola

Discussed today:

- User session flow mapping
- The content of the next merge request
- Society and event entity relationship
- Future plans. Presentation etc
- Chapter delegation for end of project report
- Sprint retrospective reformation
- End to End integration
- UX improvements
- Documentation

Meeting started: 11:15 Am
Meeting ended: 1pm

Team 11 meeting - 29th Nov 2024

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y

Nathan O'Connor	Y
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	N
Patrick Johnson Aggrey	Y
Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	My Tickets Page	Updating a few fields in the create event page	
Nathan	Start page and User story mapping	Start page and User story mapping and Logo	
Dmitrij	MyProfile page	MyProfile page	
Sam Olatilewa Ishola		Backend Home page, My feed and mytickets	Waiting on upload of MyTickets Page
Chikezie Ogbogu		Notifications and Alerts	
Patrick Johnson Aggrey		MyEvents CRUD	
Paula Catalan	Showing different interfaces to different account types	Backend for societies page	None

Scrum master: Nathan
Minutes keeper: Paula Catalan

Discussed today:

- Started meeting with update on separation of UX depending on account type, set up and running.
- Everyone should make a pull request from main so that we all have most updated version of app
- Went through backlog of last bits that need doing
- Finished dividing up report, can be found in Teams chat
- Decided on creating a logo – Nathan

Meeting started: 10:00Am
Meeting ended: 10:40Am

Team 11 meeting - 04th Dec 2024

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	Y
Dmitrijs Kraliks	Y
Sam Olatilewa Ishola	N
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam	Finished my tickets front end page, initial user registration form and the create event page.	The report	
Nathan			
Dmitrij	MyProfile - Following Others and Displaying followed profiles, displaying edit profile and create events, account details Survey Changes - how society account user profiles are followed on account launch (how this information is saved properly to work with the profile pages also) Logout and Myprofile button components for navbar Page backgrounds	MyProfile edit profile, societies long and short descriptions Account pictures for users	

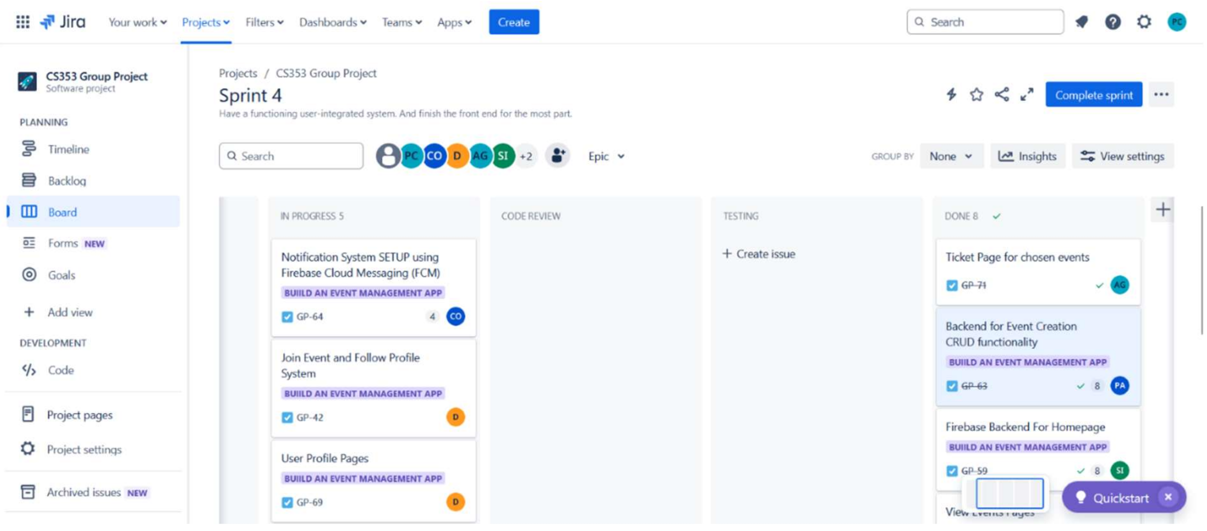
	Name field for Union on registration and general registration flow Account access links for different types of account		
Sam Olatilewa Ishola	My Feed and Home Discover Page fully integrated with backend. They are populated correctly when an event is created and display the right information Correct events are displayed based on host type filters and logged in user's personalised pages based on subscribedProfiles. All contents dynamically fetched.	User Experience Testing	
Chikezie Ogbogu	Notifications through firebase cloud based messaging	Sending Emails through the website	
Patrick Johnson Aggrey			
Paula Catalan	Back end for societies	Backend for MyTickets and for Event detail. Implement front page.	None

Scrum master: Nathan
Minutes keeper: Paula Catalan
Discussed today:

- Started meeting with small scrum meeting
- Worked on new fields to be added to collections (societies and users).
- Merged to some team members branches the newest version in main.
- Changed the CSS of the background of the app.
- Changed Maynooth uni image to personalized logo.
- Implemented nav bar in profile page.
- Discussed best UX way to use and go about profile pages when visiting other people's profiles
- Decided on way to implement image upload and download

- Started sprint 5

Meeting started: 11:00Am
Meeting ended: 13:00Am
Jira Board



Projects / CS353 Group Project

Sprint 4

Have a functioning user-integrated system. And finish the front end for the most part.

IN PROGRESS 5

- Notification System SETUP using Firebase Cloud Messaging (FCM) — BUIILD AN EVENT MANAGEMENT APP — GP-64 — 4 — CO
- Join Event and Follow Profile System — BUIILD AN EVENT MANAGEMENT APP — GP-42 — D
- User Profile Pages — BUIILD AN EVENT MANAGEMENT APP — GP-69 — D

CODE REVIEW

TESTING

DONE 8

- Ticket Page for chosen events — ~~GP-71~~ — AG
- Backend for Event Creation CRUD functionality — BUIILD AN EVENT MANAGEMENT APP — ~~GP-63~~ — 8 — PA
- Firebase Backend For Homepage — BUIILD AN EVENT MANAGEMENT APP — GP-59 — 8 — SI

<u>Team 11 meeting - 11th Dec 2024</u>

Meeting attendance

Name	Attendance
Anshuk Gaddam	Y
Nathan O'Connor	Y
Dmitrijs Kraliks	N
Sam Olatilewa Ishola	Y
Chikezie Ogbogu	Y
Patrick Johnson Aggrey	Y
Paula Catalan	Y

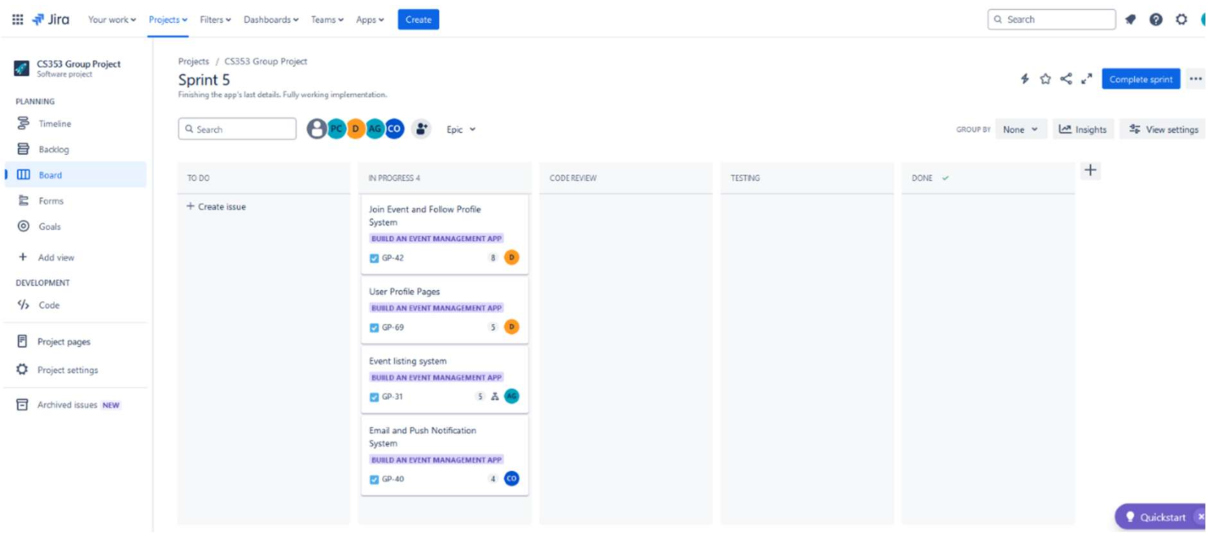
Name	Work Done	Currently Working on	Issues or blockage
Anshuk Gaddam			
Nathan			
Dmitrij	Socieities page society profile pictures	Comments and report	

	MyProfile Page functionality Bio and descriptions for students and societies		
Sam Olatilewa Ishola		Report and remote server deployment	
Chikezie Ogbogu			
Patrick Johnson Aggrey			
Paula Catalan	Finished and cleaned up the new added code i.e. EventDeatils and start page.	Comments and populating the database	None

Scrum master: Nathan
Minutes keeper: Paula Catalan
Discussed today:

- Final Meeting today
- Finalized last things needed to be done:
- Remote server upload – Sam will look into it
- Comments on code
- Finish filling in minutes "Done" table
- Chikezie's notification system will be moved into future releases
- Clarified documentation issues and doubts
- Each teammate has to upload their own report, but chapters 0,2 and 3 will be collaborative and the same.
- Word limit applies individually.

Meeting started: 11:00Am
Meeting ended: 13:00Am
Jira Board



Projects / CS353 Group Project

Sprint 5

Finishing the app's last details. Fully working implementation.

TO DO: + Create issue

IN PROGRESS 4:
- Join Event and Follow Profile System — BUILD AN EVENT MANAGEMENT APP — GP-42 — 8 — D
- User Profile Pages — BUILD AN EVENT MANAGEMENT APP — GP-69 — 5 — D
- Event listing system — BUILD AN EVENT MANAGEMENT APP — GP-31 — 5 — AG
- Email and Push Notification System — BUILD AN EVENT MANAGEMENT APP — GP-40 — 4 — CO

CODE REVIEW

TESTING

DONE ✓

UML diagrams:

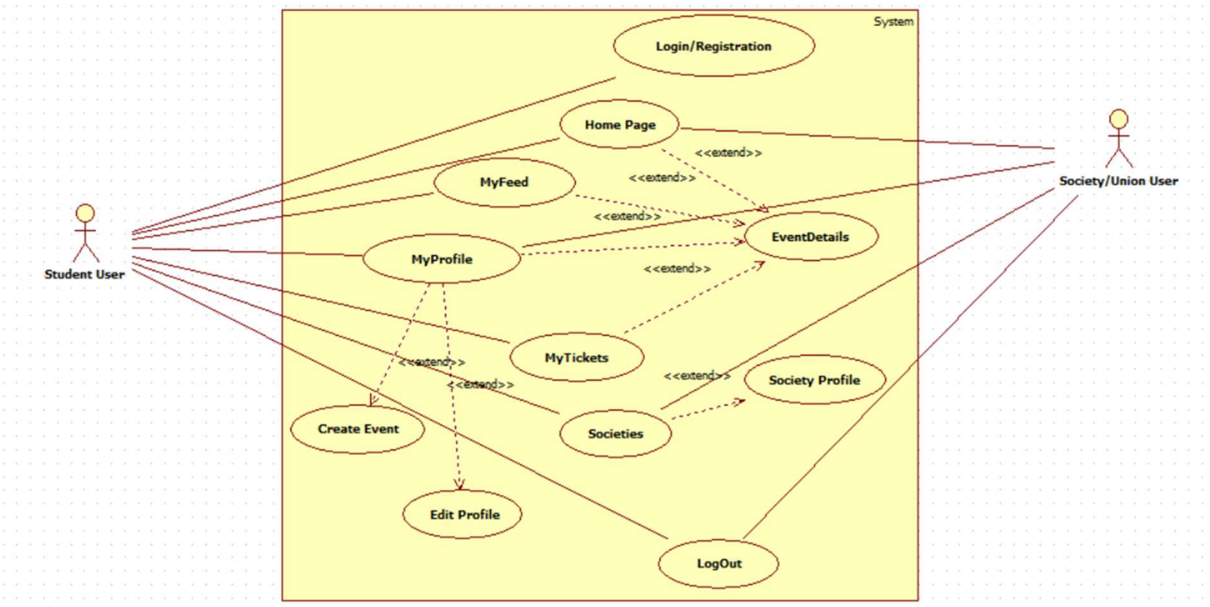


Figure 2.1: UseCase diagram

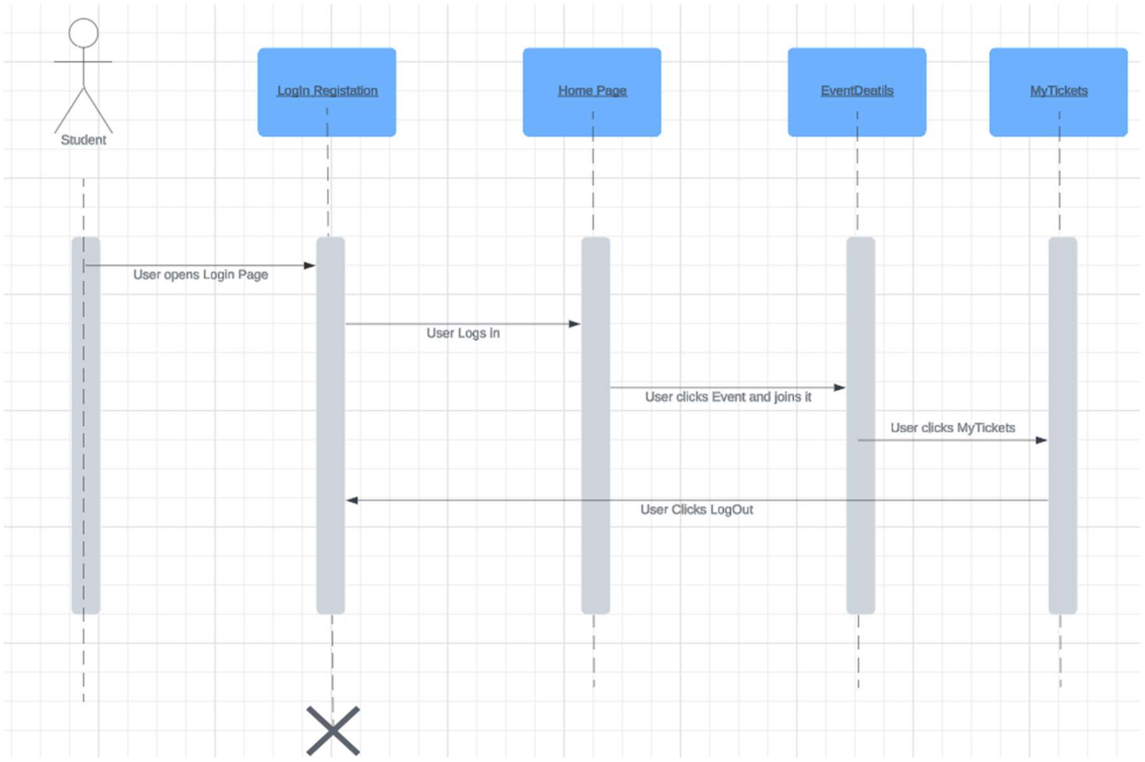


Figure 2.2: Sequence Diagram

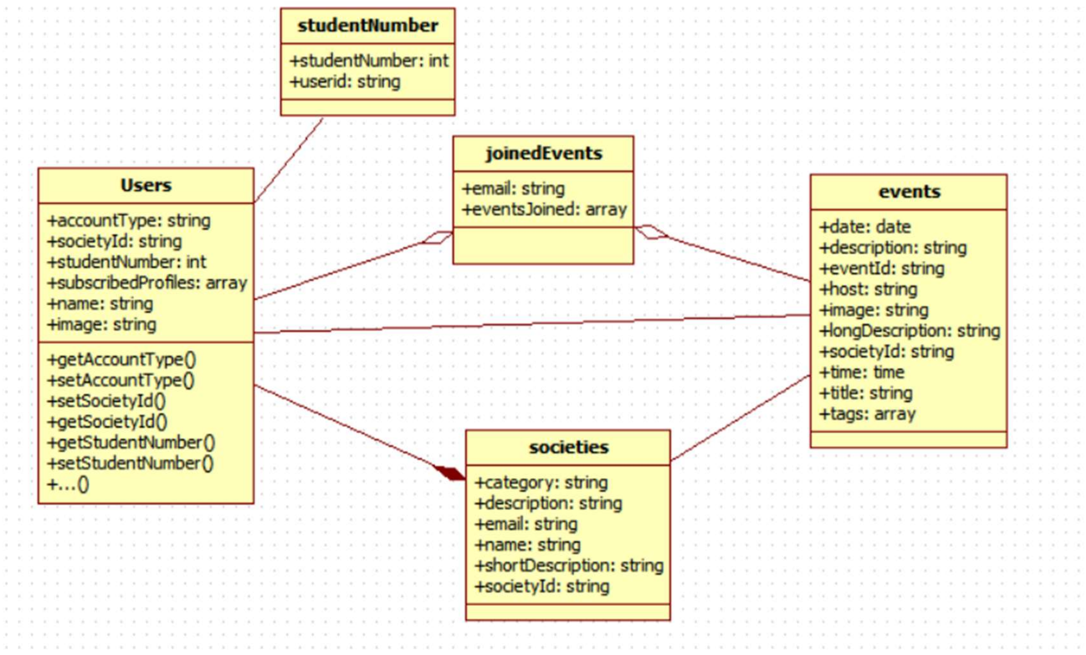


Figure 2.3: Class Diagram

Bibliography:

0.1) GeeksforGeeks, 2024. *Software Engineering | Agile Software Development*. [online] Available at: https://www.geeksforgeeks.org/software-engineering-agile-software-development/ [Accessed 20th December 2024].

0.2) Wikipedia, 2024. *Agile software development*. [online] Available at: https://en.wikipedia.org/wiki/Agile_software_development [Accessed 20th December 2024].

0.3) Kevin Casey, 2024. *CS353 – Group Project Lecture 1 – SCRUM Introduction*. [PowerPoint presentation] Maynooth University, CS353, delivered on 27th September 2024.

2.1) React, 2024. *useEffect – React Documentation*. [online] Available at: https://react.dev/reference/react/useEffect [Accessed 20th December 2024].

2.2) LogRocket, 2024. *useEffect React Hook: A Complete Guide*. [online] Available at: https://blog.logrocket.com/useeffect-react-hook-complete-guide [Accessed 20th December 2024].

2.3) Firebase, 2024. *Get Started with Firebase Authentication on Websites*. [online] Available at: https://firebase.google.com/docs/auth/web/start [Accessed 20th December 2024].

2.4) Firebase, 2024. *Cloud Firestore Quickstart*. [online] Available at: https://firebase.google.com/docs/firestore/quickstart [Accessed 20th December 2024].

4.1) OpenAI (2024) *ChatGPT*. Available at: https://openai.com/chatgpt (Accessed: Oct- Dec 2024).

4.2) Retool (n.d.) 'Filtering Data in React: filter(), map(), and for loops', *Retool Blog*. Available at: https://retool.com/blog/filtering-data-in-react-filter-map-and-for-loops (Accessed: 15th October 2024).

4.3) Mubashar, S. (2022) 'Search Bar in React JS!', *DEV Community*. Available at: https://dev.to/salehmubashar/search-bar-in-react-js-545l (Accessed 16th October 2024).

4.4) GUVI (n.d) ‘Build a Search Filter Component in React’, Available at: https://www.guvi.in/blog/build-a-search-filter-component-in-react/ (Accessed 16th October 2024)

4.5) W3Schools (n.d.) 'CSS Tutorial', *W3Schools*. Available at: https://www.w3schools.com/css/ (Accessed: Oct – Dec 2024).

4.6) Cathley, S. (n.d.) 'How to Show/Hide Modal in React', *DEV Community*. Available at: https://dev.to/cathleys/how-to-showhide-modal-in-react-46fg (Accessed: 30th October 2024).

4.7) Bankole, S. (2023) 'Firebase in React: Step-by-Step Guide', *Medium*. Available at: https://samuelbankole.medium.com/google-firebase-in-react-1acc64516788 (Accessed: 22nd October 2024)

4.8) Stack Overflow (2018) 'Is it possible to import a group of images as an array? [Create React App project]', *Stack Overflow*. Available at: https://stackoverflow.com/questions/48560592/is-it-possible-to-import-a-group-of-images-as-an-array-create-react-app-projec (Accessed: 10th December 2024).