Meer Akbar Zohair Moosavi

22925002

Meer.moosavi.2023@mumail.ie

Team 17

LyfeLine

Table of Contents

Member who posted screencast and repo: Cian Patrick Curran (cian.curran.2023@mumail.ie)

Chapter 0 – Introduction to Agile Development (SCRUM)

AGILE Introduction

AGILE development is a modern approach to project management and software development that emphasises adaptability, collaboration and delivering value. It came to light after realisations that the traditional methodologies can have some limitations. The waterfall model, a traditional method used in software development, is an example of a methodology that led to limitations. Typically, with the waterfall model, it was a linear sequence of events / planned events making it harder to change the projects trajectories or objectives once it is in motion. Making it harder to use in dynamic environments. AGILE methodologies aim to improve the speed, flexibility, and the overall quality of project delivery by focusing on iterative processes, customer collaboration, and responsiveness to change (Beck, 2001). Among these methods SCRUM is one of the most popular, offering teams a well structed yet flexible approach to managing projects.

AGILE Development: Values and Principles

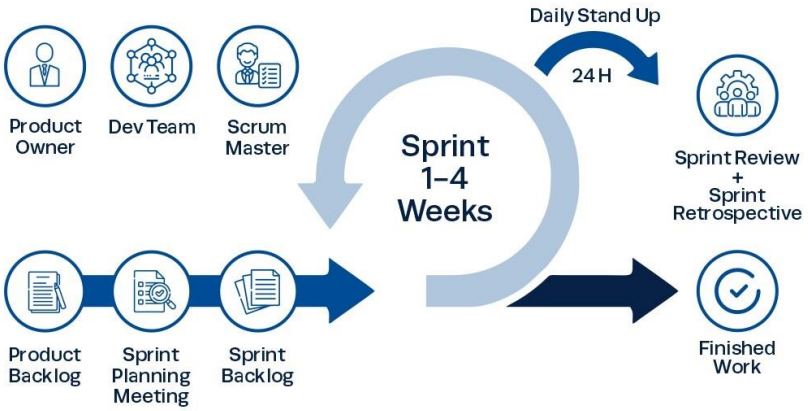
The principles of AGILE Development are rooted in the AGILE Manifesto which was introduced in 2001 by a group of software developers who were fed up with traditional methods and were seeking a more dynamic and collaborative approach to project management. Their manifesto prioritises:

1. **Individuals and interactions over Processes and Tools:** AGILE emphasises the importance of collaboration and communication within teams to ensure a successful outcome.
2. **Working Software over Comprehensive Documentation:** Delivering high level, functioning software takes priority over excessive documentation, necessary details are still included.
3. **Customer Collaboration Over Contract Negotiations:** AGILE teams work closely with the stakeholders through the entirety of the projects lifecycle to ensure the project aligns with requirements as they evolve.
4. **Responding to change over Following a plan:** AGILE embraces flexibility, allowing teams to adapt to changing circumstances and deliver consistent value (Beck, 2001).

These principles serve as the foundation for AGILE frameworks such as SCRUM, which implement them using clearing defined roles, events and artifacts.

SCRUM Framework Overview

SCRUM is a robust, lightweight framework, based on the AGILE principles, which was designed to help teams work collaboratively to tackle complex projects. The name was christened after a rugby term that signifies teamwork and coordination, SCRUM is particularly effective in fast – paced and evolving environments. It can provide teams with the tools and structure to deliver value in incremental stages while still being adaptable to change.



1 SCRUM FRAMEWORK - Institute Project Management (https://projectmanagement.ie/)

Key Roles in SCRUM

SCRUM defines three primary roles, each with their own distinct responsibilities:

1. **Product Owner:**
    - Represents the voice of the customer and ensures that the team focus is aimed towards delivering the highest value features.
    - Manages and sets prioritisation on the product backlog, a list of requirements that can constantly be changing.
    - Collaborates with the stakeholders to ensure that the product vision aligns with the business objectives.
2. **SCRUM Master:**
    - Acts as a facilitator and ensures that the team follows the SCRUM principles.
    - Remove any factors that may hinder progress and fosters an environment of collaboration.
    - Encourages continuous improvement through meetings and other events.
3. **Development Team:**
    - A group of cross functional (different skillsets) team members, typically developers and engineers, responsible for delivering the project development.
    - Self-organising and accountable for planning, executing and completing tasks within sprints.

SCRUM Events

SCRUM is built around five key events, each event designed to promote transparency, inspection, and adaptation.

1. **Sprint:**
   The core of SCRUM, a sprint is a time limited iteration (usually anywhere from one week to a month) during which the team works to deliver a potentially shippable product increment.
   Example: A development team working on a mobile application might dedicate a two week sprint to building and testing a user authentication feature, ensuring that it will be function and ready to integrate into the main application.
2. **Sprint Planning:**
   Held at the beginning of each sprint, this event involves defining the goals of this spring and selecting which items from the product backlog to work on.

Example: For a project developing an ecommerce website, a sprint planning session might focus on implementing a shopping cart feature. The team selects tasks like creating User interface designs, adding item and removal functionality, and integrating it with the checkout process.

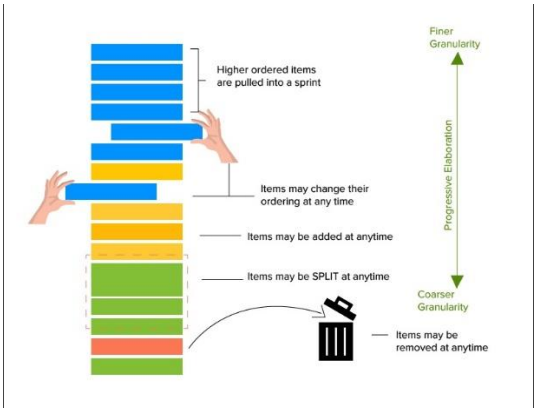
2 SPRINT planning - https://khatrishashank.wordpress.com/

3. **Daily SCRUM:**
   A short 10–15-minute meeting where progress is discussed within the team, plans for the day, and identify any bugs or issues that could be encountered.
4. **Sprint Review:**
   Conducted at the end of the sprint, this event allows the team to highlight the completed work (or what they have developed so far) to the stakeholders.
   Example: During the review, the team demonstrates the newly developed shopping cart functionality to the stakeholders. Feedback is gathered, such as adding a "Save for Later" option or making the cart UI visually consistent with the rest of the site.
5. **Spring Retrospective:**
   A reflective meeting where the team evaluates their performance and identifies areas of improvement.

SCRUM Artefacts

SCRUM relies on three main artifacts to provide visibility and maintain focus:

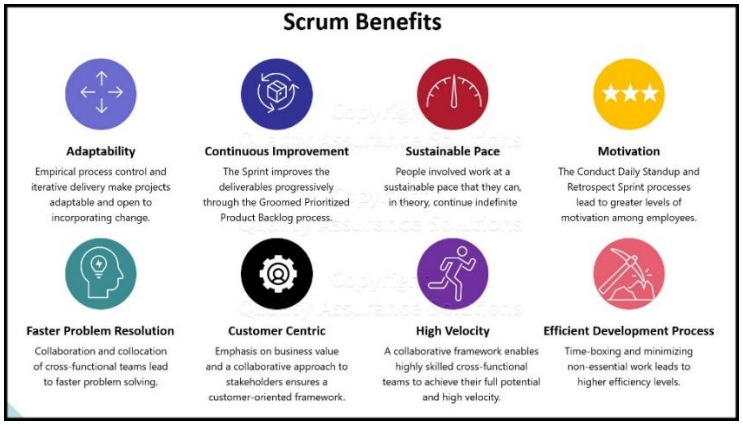
1. **Product Backlog:**
   - A dynamic, prioritised list of all features, enhancements, and bug fixes for the product.
   - Mainly managed by the product owner to ensure alignment with business goals.



3 Working with the Scrum Product Backlog (https://www.scrum-institute.org/The_Scrum_Product_Backlog.php)

2. **Sprint Backlog:**
    - A subset of the Product backlog selected for the current sprint, along with a plan to achieve the sprint goal.
3. **Increment:**
    - The sum of all completed product backlog items during a sprint, representing a step towards the final product.

Benefits of SCRUM



4 The Scrum Tool - Benefits of Scrum (www.quality-assurance-solutions.com/scrum-tool.html)

SCRUM offers several advantages, making it a popular choice for AGILE teams:

1. **Enhanced Collaboration:**
    - Regular events such as the Daily SCRUM and sprint reviews promote communication and teamwork.
2. **Increased Flexibility:**
    - The iterative nature of sprints allows teams to adapt to changing requirements without disrupting the overall project or the flow of production.
3. **Faster Delivery:**
    - By focusing on delivering small, functional increments, teams can provide value to customers much quicker. It also allows multiple teams to work on different increments at the same time.
4. **Improved Quality:**
    - Regular reviews encourage teams to continuously improve their processes and deliverables.
5. **Higher Transparency:**

- The continuous meetings ensure that everyone involved will have a clear understanding of what their objectives are, as well as ensuring that the stakeholders can track development to ensure its meeting their business objectives.

Challenges with implementing SCRUM

Despite its benefits, implementing SCRUM can have its challenges and downsides:

1. **Cultural Resistance:**
    - Teams transitioning from more traditional methods may struggle to embrace the self-organising nature of SCRUM.
2. **Role Misunderstandings:**
    - Misalignment or confusion about the responsibilities of the Product Owner or SCRUM master can lead to inefficiencies.
3. **Overcommitment / under commitment:**
    - Teams may overestimate their capacity, leading to incomplete sprints and frustration. Teams that under commit may run into the same issues and could lead to incomplete sprints.
4. **Dependency Management:**
    - Coordination with external teams or dependencies outside of the sprint can disrupt progress.
5. **Time Management:**
    - Adhering to time boxed events requires discipline and focus, especially for teams that are new to using the SCRUM method.

Scrum in Practice

SCRUM has been widely adopted across industries beyond software development, including marketing, education, and civil service. Tools such as Mira and Trello are often used to manage SCRUM processes. This enables the teams to track progress and maintain full transparency. A successful SCRUM team will demonstrate:

- Continuous learning through retrospectives.
- Effective communication between stakeholders and development teams.
- Adapting Practices to suit the unique needs of the team and project.
- Good time management skills to ensure that goals are met in time boxed events.

Conclusion

SCRUM is a versatile and powerful framework for AGILE development, providing a structured approach to delivering complex projects in increments. By fostering collaboration, adaptability, and continuous improvement, SCRUM allows teams to respond effectively to changing requirements and deliver high-quality products. While its implementation can pose some challenges, the benefits of increased Flexibility, faster delivery, and enhanced transparency make it a valuable methodology in the modern project management world.

Chapter 1 – My Proposal:

When I submitted my project proposal, my goal was to create a platform aimed at improving healthcare management. Specifically, I envisioned developing a system to assist doctors and patients in managing prescriptions, appointments, and treatment plans more efficiently. The platform would feature tools like

uploading prescriptions, tracking medications, setting reminders, and facilitating communication between doctors and patients.

The idea stemmed from my passion for using technology to address real-world challenges. I believed this project could provide a meaningful application of my technical skills while addressing critical needs in healthcare. My plan was to collaborate with medical professionals to ensure the system was practical and relevant to their requirements.

Rejection and the Need to Reassess

Unfortunately, my proposal was rejected during the review process. The panel's feedback highlighted that my project, while innovative, did not align with the scope of the course. They explained that the complexity of the proposed idea, including the need for collaboration with healthcare professionals and compliance with data privacy regulations, exceeded the course's time and resource constraints.

This feedback was disappointing, as I had invested significant effort into shaping the proposal. However, it became clear that I needed to adjust my approach and redefine my goals to focus on something more feasible within the limitations of the course.

Shifting Focus to a Realistic Goal

In response to the feedback, I decided to revise my proposal and focus on a more manageable project. I shifted from a healthcare management platform to a library management system—a simpler yet meaningful idea that allowed me to explore my interest in data management and technology.

The revised project involved building a system to help users manage books, track reading progress, and interact with features like searching for books, saving them for later, and updating their reading status. This pivot allowed me to concentrate on web development, database management, and user interface design—core technical skills that aligned with the course's objectives.

Why the Revised Proposal Was Accepted

The revised proposal was accepted because it addressed the concerns raised in the initial review. It was more practical and achievable, eliminating the need for external collaboration and regulatory compliance. By narrowing the scope, I could focus on delivering a high-quality project within the given timeframe.

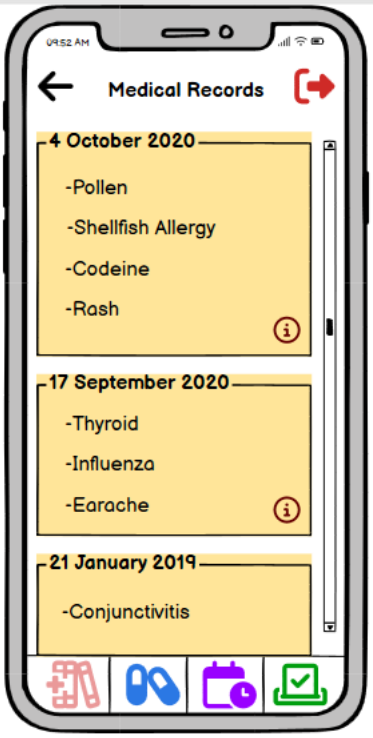
Key features of the library management system included a search bar, a dynamic book status tracker, and user-friendly design elements. These components demonstrated my ability to build functional and engaging systems while honing my technical skills.

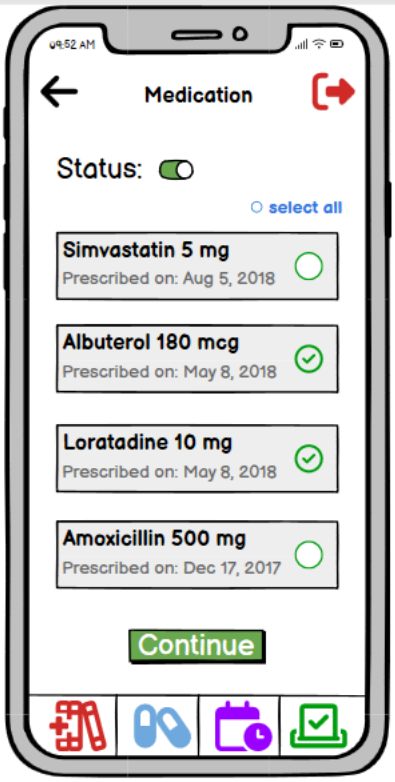
Moving Forward: Lessons Learned

This experience taught me critical lessons about adaptability, project planning, and the importance of setting realistic goals. While I remain passionate about using technology to address real-world problems, I now understand the need to break ambitious ideas into achievable steps.

The library management system project, though simpler than my initial vision, allowed me to develop a strong foundation in web development, database management, and user interface design. These skills will undoubtedly benefit me in future projects, whether in healthcare or other fields.

In the end, the rejection of my original proposal was a stepping stone toward understanding the balance between ambition and feasibility. Moving forward, I'm better prepared to approach projects with a realistic and strategic mindset, ensuring that my efforts lead to meaningful and achievable outcomes.




Medication
Status:
select all
Simvastatin 5 mg
Prescribed on: Aug 5, 2018
Albuterol 180 mcg
Prescribed on: May 8, 2018
Loratadine 10 mg
Prescribed on: May 8, 2018
Amoxicillin 500 mg
Prescribed on: Dec 17, 2017
Continue

Chapter 2 – The actual project

Our project, **LyfeLine**, was born from the collective creativity and collaboration of our team during the early brainstorming phase. Rather than focusing on a single proposal, we merged the strengths of all our initial ideas into one unified platform designed to improve the day-to-day lives of its users. The name *LyfeLine* reflects on its purpose—acting as a reliable, multi-functional hub to support individuals in managing everyday challenges effectively.

While LyfeLine was designed with students in mind, it has a broader appeal and can benefit anyone, regardless of their background. For students, life can often feel overwhelming when juggling studies, traveling, finances, and schedules, and LyfeLine aims to address these challenges through its intuitive and practical features. For example:

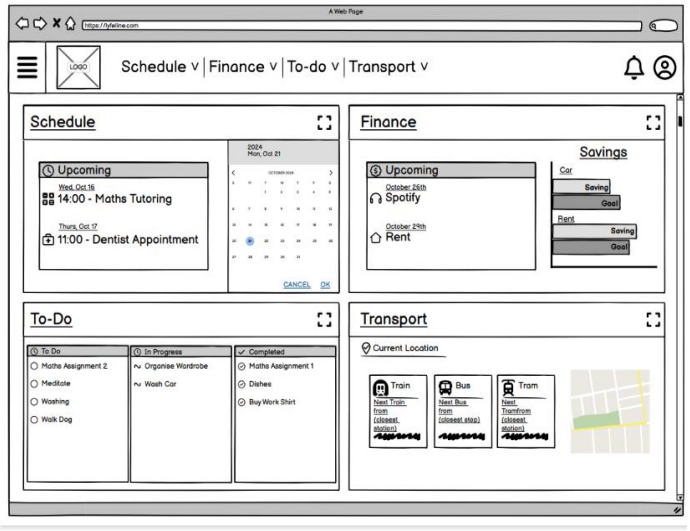
- **Task Management**: The built-in to-do list helps students stay organised with assignments, deadlines, and personal goals, ensuring they can balance study and other commitments.
- **Commuting Assistance**: Geolocation tools track **bus schedules** (Using a real time API from Transport For Ireland) and commuting routes to and from college, saving time and reducing stress for students navigating their daily journeys.
- **Financial Management**: Managing finances is particularly tough for students balancing part-time jobs, budgeting for expenses, and saving for future goals. LyfeLine's finance tools provide a clear way to **track income, expenses, and savings goals**, making it easier to stay in control.
- **Library and Resources**: The library section offers quick access to educational and organisational resources, providing support for learning and personal development.

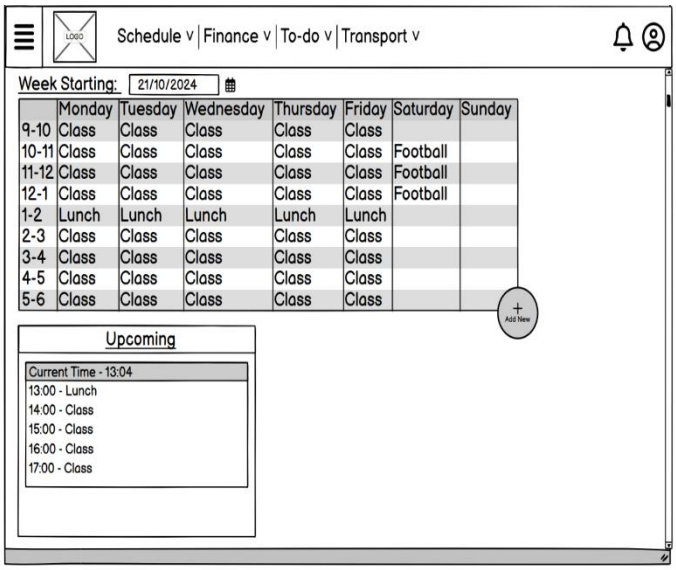
Despite its focus on helping students, the platform's tools are versatile enough to benefit a wider audience, including working professionals, busy parents, and self-employed individuals looking to streamline their daily lives.

Development and Design

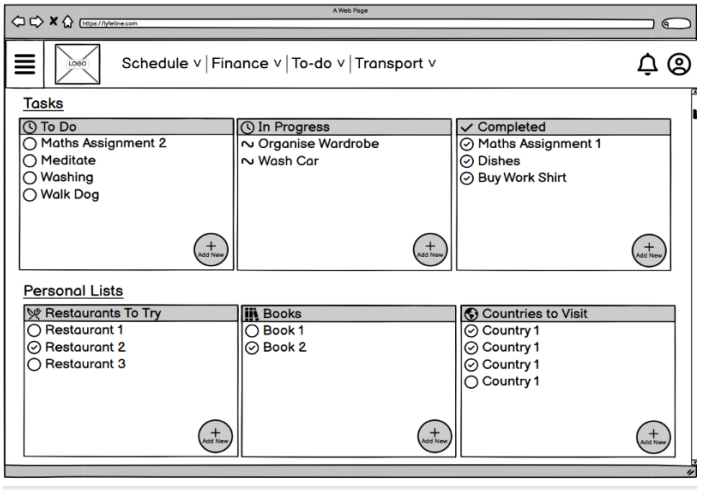
During the initial phases, our team worked collaboratively to ensure **LyfeLine** would be both practical and user-friendly. By Week 2, we adopted a structured approach to visualising the project. Each team member was tasked with creating wireframes for their respective sections—finance tools, transport tracking, to-do lists, schedules, and the library. This step allowed each team member to contribute meaningfully to the project, ensuring everyone's vision and ideas were represented.

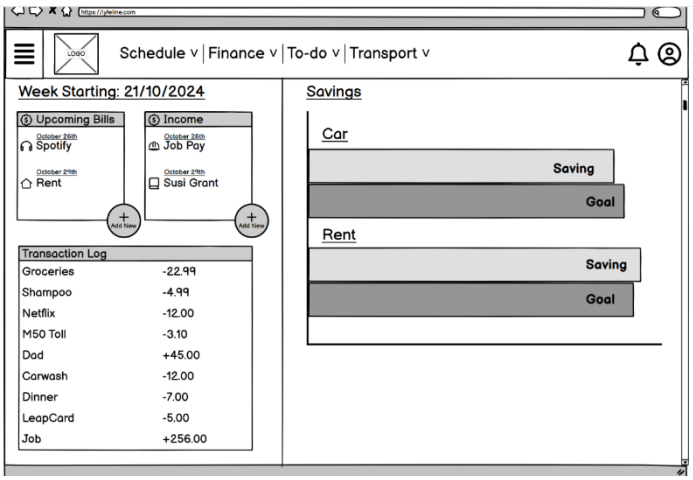
The wireframing process was critical in shaping LyfeLine's design. As a team, we compared and analysed each design to identify common themes, focusing on usability, simplicity, and visual consistency. By combining our insights, we selected a unified design that was clean, intuitive, and accessible for all users. This cohesive approach ensured that every feature seamlessly fit into the overall user experience.

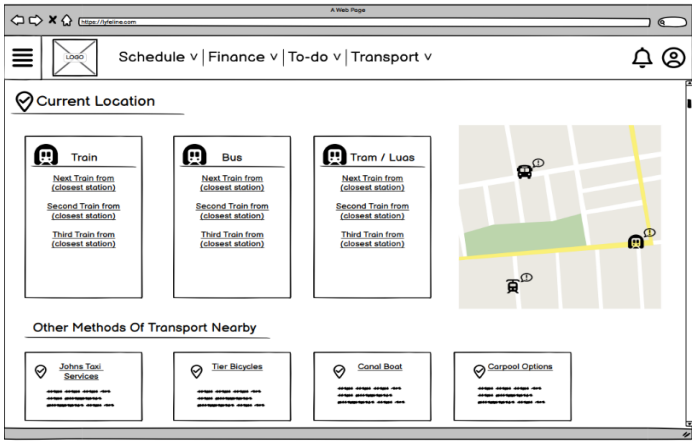




	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9-10	Class	Class	Class	Class	Class		
10-11	Class	Class	Class	Class	Class	Football	
11-12	Class	Class	Class	Class	Class	Football	
12-1	Class	Class	Class	Class	Class	Football	
1-2	Lunch	Lunch	Lunch	Lunch	Lunch		
2-3	Class	Class	Class	Class	Class		
3-4	Class	Class	Class	Class	Class		
4-5	Class	Class	Class	Class	Class		
5-6	Class	Class	Class	Class	Class		







The wireframes presented here were created by Alex, whose designs were unanimously recognised by the group as the best representation of our collective vision for the LyfeLine project. These wireframes effectively captured the core concepts we had discussed, striking the perfect balance between usability, simplicity, and visual clarity.

What stood out most was how these wireframes seamlessly integrated all of LyfeLine's key features—such as task management, transport tracking, financial tools, and scheduling—into an intuitive and user-friendly layout. The clean design and logical flow ensured that each section felt connected while still maintaining its individual purpose, offering users a cohesive experience.

While the wireframes served as a baseline and guide for the project's development, it's important to note that not all the features shown in the initial designs made it to the final version. For example, the scheduling tool—while valuable—was ultimately set aside as we focused on refining the core features that best aligned with our goals and time constraints. If time allowed, we would be adding this feature in at a later date.

Nonetheless, the wireframes provided a strong foundation and clear direction for the team. They not only reflected our collaborative efforts but also laid the groundwork for the final design of LyfeLine, ensuring the platform delivered a practical and polished experience for its users.

A Unified Vision

The collaborative process of merging ideas, creating wireframes, and refining our design enabled us to develop a platform that reflects the diverse needs of its users. **LyfeLine** isn't just a website—it's a tool designed to add value to everyday life, whether you're a student navigating your studies, a professional managing work tasks, or someone simply looking for a better way to organise your day.

By combining practicality with creativity, our team created a product that can act as a true “lifeline” for anyone seeking to simplify and improve their daily routine.

Technologies/architecture used

Once we had our website idea around week 2/3, we started to discuss what technologies and platforms we would use to code out project. We agreed quickly that we all wanted to use VSC (Visual Studio Code) as we felt it would be the easiest to code and to manage. We also said we might try experiment with react. We all did some research and practice to learn it but most of the group ended up not using it. I however did so all my frontend code was in HTML, CSS and JavaScript (with react) files. For all my backend code then I used Node.js. I used MinGW to help my coding as well. We also agreed that if we got to a stage where we were incorporating databases to save information to the website, we would do this on PGadmin as we all had previous experience with PGadmin. I used PGadmin a lot and managed to get a background database running to my to-do list which would use push and delete commands to save and delete information to the to-do list. Around sprint 4 we also decided we would get GITLAB working. So, Alex McGrath and I spent some time getting a repository set up. We set up the repository and then I uploaded everyone’s front end files that Joanna had uploaded when she combined all the front end files. Around sprint 5 is when I was sent all the backend files and then uploaded them to GIT. GIT turned out to be extremely useful as everyone now had access to a central location for our file management.

Client-Server Model:

- The project follows a two-tier architecture where the frontend (client) communicates with the backend (server) via RESTful APIs.

Frontend: The front end is developed using a combination of technologies, including react for specific components and plain HTML, CSS and JavaScript for other parts.

Example of our front-end code

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>LyfeLine - Library</title>
    <link rel="stylesheet" href="./Library.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">
</head>
<body>
    <div class="navbar" id="navbar">
        <div class="nav-section">
            <a href="studypage.html">Study</a>
            <a href="Library.html">Library</a>
            <a href="finance.html">Finance</a>
            <a href="todolist.html">To-Do</a>
            <a href="transport.html">Transport</a>
        </div>
        <div class="logo-section">
            <a href="HomePage.html">
                <img src="assets/logo.png" alt="Logo" class="logo">
            </a>
        </div>
        <div class="profile-section">
            <a href="accountpage.html">
                <img src="assets/userIcon.png" alt="Profile" class="profile-icon">
            </a>
        </div>
    </div>

    <div class="search-section">
        <input type="text" id="search-bar" placeholder="Search for a book...">
        <select id="filter-dropdown" class="filter-dropdown" onchange="filterBooks()">
            <option value="">All</option>
            <option value="read">Read</option>
            <option value="reading">Reading</option>
            <option value="awaiting">Awaiting</option>
            <option value="bookmarked">Bookmarked</option>
        </select>
        <button class="search-button" onclick="searchBooks()">Search</button>
    </div>

    <div class="book-display">
        <h2>Book Collection</h2>
        <ul id="book-list">
            <!-- Books will be populated from database -->
        </ul>
    </div>

    <script src="./Library.js"></script>
</body>
</html>
```

Backend:

- The backend contains separate routes for handling CRUD operations related to the To-Do List and database communication.

Example of backend code where the user is connecting to server.

```
const express = require('express');
const { Client } = require('pg');
const path = require('path');

const app = express();
app.use(express.json());
app.use(express.static(__dirname));

const client = new Client({
    host: 'localhost',
    database: 'Lyfeline_db',
    user: 'postgres',
    password: 'Group17',
    port: 5432,
});

client.connect(err => {
    if (err) {
        console.error('Connection error', err.stack
    } else {
        console.log('Connected to database');
    }
});
```

Database Schema:

- Tasks: Table for storing task details (e.g., ID, title, description, due date).

Flow:

- **User Interaction**: The user interacts with the React-based UI.
- **API Requests**: The frontend sends API calls (e.g., GET, POST) to the backend.
- **Database Operations**: The backend interacts with MySQL to fetch, insert, update, or delete data.

Chapter 3: The SCRUM Process

Initial Meeting

From the very beginning, we felt that our group demonstrated a strong ability to conduct SCRUM meetings effectively. We didn't fully understand what the SCRUM process was initially, but we grasped it very quickly. Our team was composed of highly motivated individuals who were not only eager to contribute their own ideas but also willing to support each other in meaningful ways. This collaborative mindset created a positive and productive atmosphere, which greatly contributed to the success of our SCRUM meetings. As a result, these meetings were characterised by open and effective communication, where everyone had the opportunity to express their thoughts and ideas. Our initial SCRUM meeting set the tone for the rest of our project. Dyring this meeting each member had the chance to pitch their ideas. These pitches included not only the basic concepts but also explanations of why we believed our ideas were valuable and how they could contribute to the project's goals.

Every suggestion pitched was met with enthusiasm and sparked genuine interest across the group and it was the same for every other proposed idea. It was clear that everyone was actively listening, and there was a sense that each contribution was being seriously considered. This level of engagement made the meeting feel inclusive and rewarding for everyone involved. Once all the ideas had been presented, we transitioned into a productive discussion about which direction to take. Each concept was evaluated thoroughly, and the conversation was marked by thoughtful input from all team members. Joanna then proposed a creative and innovative solution: instead of choosing just one idea, we could combine all of them into a single, unified concept. This approach was seen as both practical and visionary, and the team quickly reached a consensus to move forward with this strategy.

We were particularly pleased with how smoothly and efficiently the first meeting unfolded. Despite the diversity of team ideas and the potential for differing opinion, we were able to reach a decision quickly and effectively as a group. This demonstrated not only the strong communication skills within our team but also a shared commitment to collaboration and progress. Reflecting on this experience, we feel it set a solid foundation for the rest of our work together, showcasing our ability to tackle challenges and make decisions about a cohesive unit.

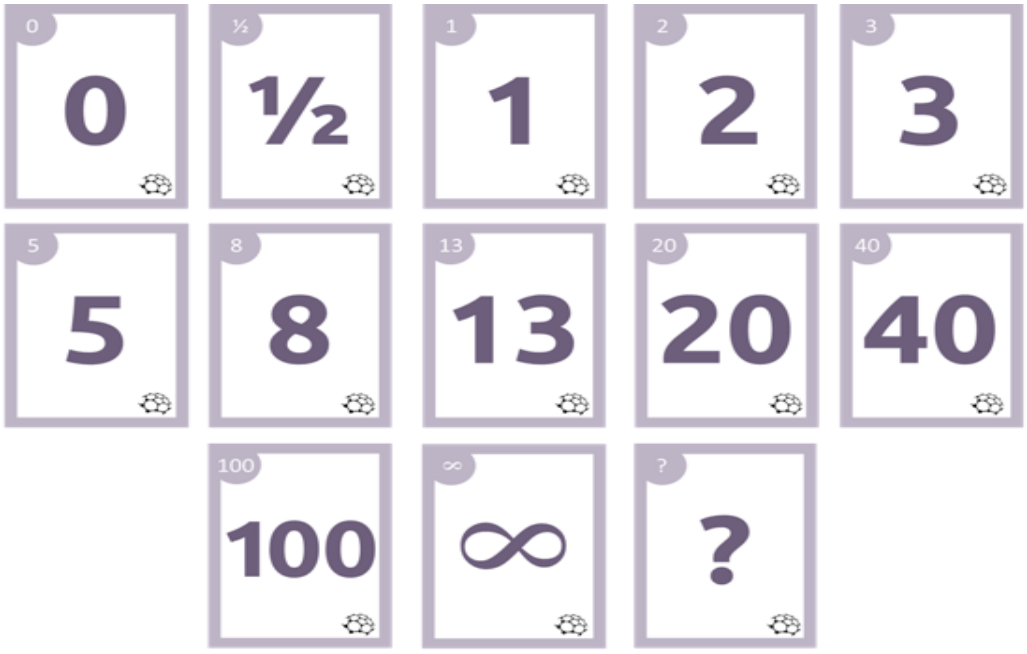
In terms of sprints, they were planned every second week. We found this to be a very efficient amount of time as it helped our team to split up our work based on these sprints. For example, allocating a certain number of sprints to frontend, backend, testing etc. We knew from the very first meeting that we would stick to these SPRINT timelines and how they would effectively help us manage our workload.

Through research of the SCRUM process and Sprints, the above graphic is the rough idea as to how we undertook all our sprints and SCRUM meetings.

Project Estimation

We began our project estimation and planning poker during the first week of our initial SPRINT. We divided the project into three main parts: front end, back end, and integration. We collectively agreed that integration would be the most challenging aspect. Our timeline allocated Sprint 1 for front-end development, Sprint 2 and the first week of Sprint 3 back-end development, The second week of Sprint 3 and the entirety of Sprint 4 and 5 for integrating the front end with the back end. Throughout the integration process, we were also continuously testing to ensure the product was working as intended. We adhered strictly to this timeline, which we believe was a key factor in our success as a team.

For our planning poker, we used the following cards and assignments.



Code Work Assignment:2

- This is where we split up who was going to code each part of our project. We felt this only

deserved a 2 as it would be very easy to do and would take minimal time.

Wireframe Creation:8

- We felt wireframe creation was an easy enough task to undertake. We knew that again it would be very easy however; we also knew it would take a fair bit longer than the code work assignment. We gave around a week for all of us to upload our wireframes which is why we gave it an 8.

Front end:20

- This is the first big task we assignment planning poker cards to. We felt that front end development was very important, and we knew it would take a good amount of time which was why we gave it a 20. We assigned around 2 weeks for it.

Testing:20

- Testing was another job we felt was quite big. We tested continuously mainly throughout the integration process. We felt testing was very important and knew it would take up some time, so we assigned it a 20 also.

Report Preparation:20

- We initially didn't think the report would take as much time as it did, but we very quickly realised it was a lot bigger than we realised. We first assigned the report an 8 but changed it to a 20 when realised it would take multiple weeks to complete.

Back end:40

- We knew very early on that back end development would take up a big chunk of our project and would be a lot more difficult than front end. For this reason, we gave it a 40.

Incorporating Front End and Back End Running:100

- Similar to back end development, we knew the incorporation of the front end and back end sections would take the longest out of anything. We knew that getting everyone's code and making it all work together would be the most difficult part. This is why we gave it the hardest score of 100.

Minutes

Regarding minutes, we experienced some challenged as a group, particularly with a few minutes' files being misplaced. However, we believe we were successful in delegating the responsibility of taking minutes. Each member took turns recording the minutes at different stages of the project, ensuring equal distribution of this task throughout the semester. While the quality of the minutes varied from week to week, we feel that many of them were well documented and contributed positively to our overall process. Rory also pitched the idea a few weeks into the project to put together a minute's file. This file was essentially just for file management and made it, so everybody had access to the documented minutes for every meeting that had happened. We formatted every minute's file as Minutes_(dd/mm/yyyy). This greatly helped when it came to writing our report. It helped in 2 aspects, helping us look back and remember what we did each week, and helped with the actual documentation of our minutes that we put into our project.

The following is the minute sample we were basing ours off, we just felt not all elements of it were necessary.

SAMPLE BUSINESS MEETING MINUTES FOR JONES CONSULTANTS, INC.

I. MEETING DETAILS

Chairperson: Kathleen Jones
Secretary: Andrew Cruz

Date: January 11, 2021
Time: 10:00 am

Location: Jones Consultants' Headquarters
Street Address: 1234 Main St.
City: Los Angeles
State: CA Zip: 91111

II. ATTENDEES.

- Kathleen Jones
- Andrew Cruz
- Jane Richardson
- Matthew Johnson
- Caroline Vogel
- Brady Foster

III. ABSENCES.

- Jeffrey Bowler
- Sarah Andrews
- Clayton Pelch

IV. CALL TO ORDER.

Previous Meeting Minutes - Speaker Kathleen Jones - 10:05 am

There was a call to order by the Chairperson to approve the meeting minutes from the January 4, 2021 meeting. Attendees voted unanimously to approve the minutes.

Current Meeting Agenda - Speaker Kathleen Jones - 10:15 am

The meeting agenda was distributed to all attendees, who were given time to review for any questions or additions.

At 10:23 am, the attendees voted unanimously to approve the agenda without any additions.

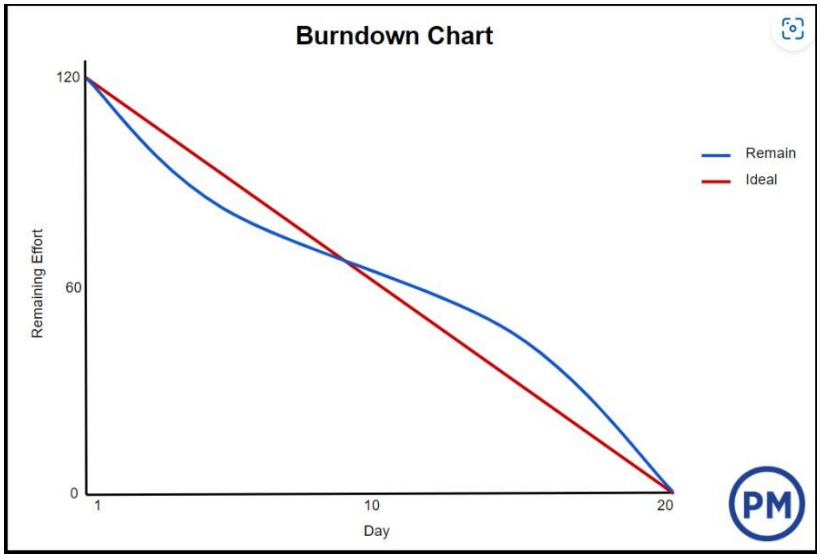
V. OLD BUSINESS.

Search for New Office Manager - Speaker Jane Richardson - 10:24 am

Page 1 of 2

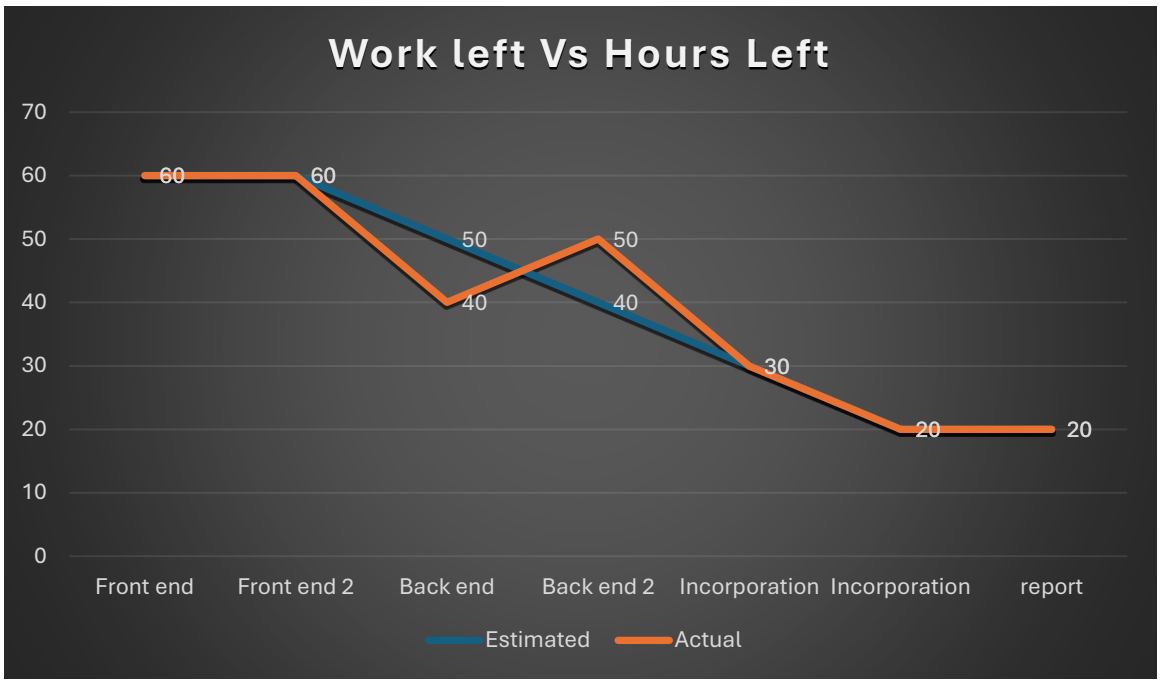
Burndown chart

A burndown chat is a simple graph used in project management to track how much work is left to do versus how much time is left to finish it. It helps teams see if they are on track to meet their goals.



Sample Burndown chart

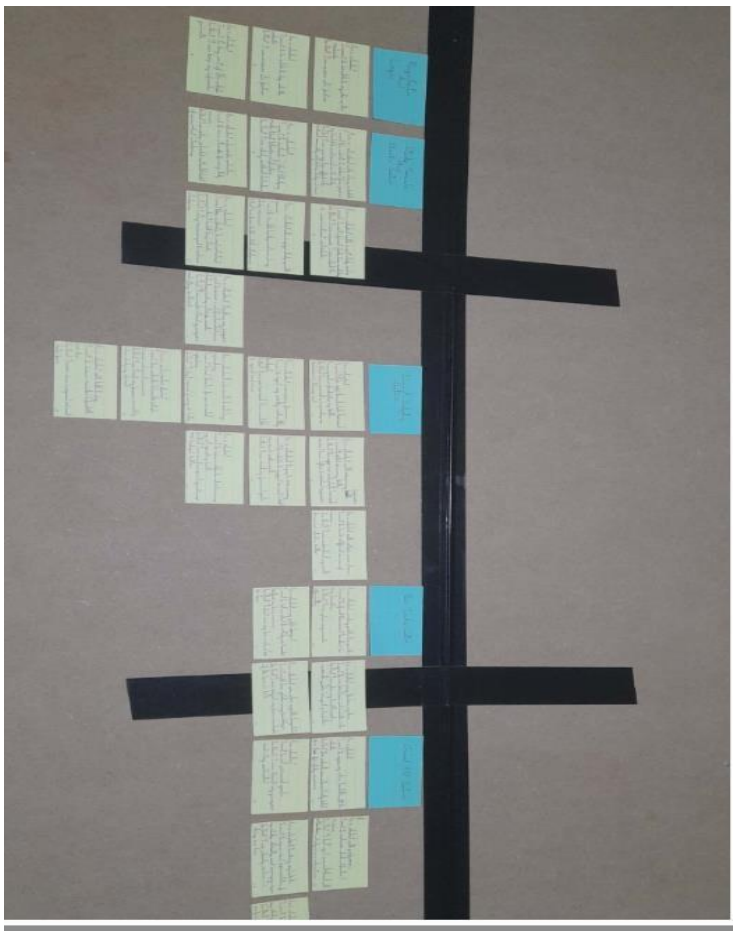
We made our burndown chart in week 2. We based it mainly off our planning poker and we felt it was a good representation of how we wanted the project to go. Our X-Axis represents ours left, while our Y-Axis represents the tasks, we undertook in the order we did it.



We started off strong originally, keeping on track for front end, however when we got on to the back-end section, we kind of fell behind as we didn't get it fully done. But towards the end of our back end development, we managed to get a lot of work done and actually managed to get ahead of schedule and start incorporating before we needed to be. When it came to the incorporation, we fell back into our estimated timeline and got everything we wanted to incorporated.

User Stories/Story Maps

Our plan and deadlines for our story maps worked well. We gave approximately a week for everyone to properly Invision the website and what they expected to be able to do with it. Everyone had their story maps uploaded either the following Wednesday (which was our deadline) or very shortly after. We all analysed all the user stories and story maps together and we all felt that Cian's were definitely the best. For this reason, we used his as our main guide. Cian's story maps are shown below. For the rest see appendix.



Testing

When building an app, the planning phase is important, but the testing phase is just as important as we need to be sure that everything is running smoothly. This phase was the most fulfilling by our members as we got to see how all our individual hard work got put together and how each part worked in one application. Testing also allowed us to find bugs that were not visible when we were working individually and make important amendments to the code. During the coding phase, we had multiple versions of code that were being updated regularly as there would be changes to the project as the weeks went by. With every version of code being changed meant that our project was constantly being improved and made to be more efficient.

There was a situation during the code joining phase that made us realise how important the testing phase is when you are building a full stack web application. Joanna, who was in charge of combining the team's code together, encountered a problem with Presley's code. The code that Presley submitted was not working well with the rest of the team's code. This was very weird as Presley tested the code on several devices to make sure that it was working before giving it to Joanna. Fortunately, she was able to identify the problem and successfully inserted Presley's code into the project without much hassle. If Joanna didn't use her due diligence to test the project, she wouldn't have identified the issues needed to be resolved meaning we would have been forced to present a partially working application.

Future releases/sprints

We also evaluated the LyfeLine app project and considered how it could evolve in future releases. With additional time, we could have implemented many more features to enhance the app's functionality and user experience. The project met our expectations within the time constraints, and we were pleased with the final outcome. However, due to the tight timeline, some ideas had to be shelved, as they were either complex or required significant time to develop.

Each team member contributed valuable suggestions that, if implemented, could have drastically improved the app's functionality. For example, features such as advanced budget analytics or enhanced study streak tracking were discussed but ultimately deferred due to time limitations. While these additions would have been beneficial, they also carried potential challenges and risks that needed more time to address effectively.

If we had more time to iterate, we could have made substantial progress toward incorporating these features, bringing the app closer to our vision. Despite this, we are proud of what we achieved within the given timeframe. The LyfeLine app ended up in a strong position, and it provides a solid foundation for future development.

Code versioning

Code versioning was an essential part of our LyfeLine group project, as it allowed the team to collaborate effectively on the development process. Using Git, we managed our codebase with ease, enabling multiple members to work on different features simultaneously without overwriting each other's work.

The use of version control tools ensured that every change was tracked, allowing us to revert to previous versions if issues arose. This was especially helpful when testing new features, as it minimized the risk of losing stable code. We also used branching strategies to separate development tasks, making it easier to integrate updates and maintain the integrity of the project.

GitHub

Team Communication/management

Group management and communication for the LyfeLine app project had its ups and downs, but overall, it was effective. There were moments of miscommunication that caused some confusion and minor delays, but these issues were resolved through team discussions, ensuring the project stayed on track. To evenly distribute responsibilities, we implemented a rotating leadership system, where each member took turns acting as the group leader or Scrum Master for a week. This approach not only lightened the burden on any one individual but also gave everyone valuable leadership experience for future projects.

While communication could have been more consistent at times, the use of tools such as Teams channels significantly improved collaboration. This allowed team members to share updates, ask questions, and exchange links to files and documentation efficiently. Regular in-person meetings also played a crucial role in aligning our efforts and clarifying each person's tasks, ensuring we made steady progress on the project.

Despite occasional challenges, our teamwork and dedication enabled us to complete the project successfully. Each group member stepped up when it mattered, contributing to the final outcome. Although there were areas where management and communication could have been improved, these experiences provided important lessons that strengthened our ability to work as a team. Ultimately, we accomplished our goals and were proud to present our project to our peers.

Remote Working

Remote working played a significant role in the development of our LyfeLine group project. Leveraging tools like GitLab, Teams, and Google Meets, we effectively coordinated tasks and maintained consistent communication despite not always being in the same physical location. Regular virtual meetings ensured everyone was aligned on objectives, while file-sharing platforms allowed for seamless collaboration on code and documentation.

Working remotely required strong time management and clear communication to avoid misunderstandings and keep the project moving forward. Despite the occasional challenge, such as scheduling across different availability, our team adapted and succeeded in maintaining progress. Remote working ultimately demonstrated the importance of flexibility and proactive collaboration, both of which were instrumental in delivering the final product.

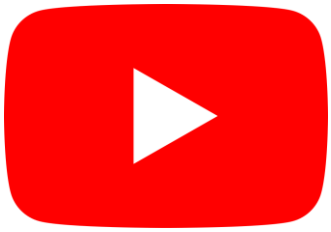
AI Experience

The use of AI tools and YouTube videos played a significant role in the development of our LyfeLine group

project. AI tools, such as ChatGPT, were essential for problem-solving, brainstorming ideas, and providing guidance on coding challenges. Whether it was understanding complex algorithms, optimizing code, or generating solutions for bugs, AI offered quick and reliable support, helping us save time and improve productivity.

YouTube tutorials were equally beneficial as they provided visual and step-by-step guidance for specific development tasks. For example, we relied on videos to learn best practices for implementing features like frontend design, user authentication, and integrating APIs. These tutorials were especially useful when we encountered unfamiliar tools or frameworks, as they allowed us to bridge knowledge gaps effectively.

By combining AI assistance with educational resources from YouTube, we were able to overcome challenges, expand our technical skills, and deliver a polished final product. This approach highlighted the importance of leveraging modern learning tools to enhance collaboration and problem-solving within the project



Chapter 4: My Contribution

4.1 Background Research

At the start of the project, I dedicated significant time to background research to guide our application's design and functionality. This included:

1. **User Experience Design**: I delved into design principles that prioritize usability, ensuring our application would be intuitive and accessible for students. This involved studying visual hierarchy, responsive layouts, and accessibility standards.
2. **Feature Analysis**: To shape our project's direction, I examined popular productivity tools tailored for students. This analysis inspired features like the 'Library Page' and the introduction of a 'Study Tracker,' ensuring they aligned with real-world needs.
3. **Technical Exploration**: I assessed various front-end and back-end technologies to identify the most suitable options. My research helped the team choose PostgreSQL as our database solution for its reliability and scalability.

4.2 Key Contributions

My core responsibilities revolved around developing the **Library Page** and related functionalities, as well as contributing to the backend and overall project infrastructure. Here's an outline of my contributions:

4.2.1 Backend Integration

- I set up the backend for the project using Node.js, connecting all pages to ensure they worked cohesively. This included interconnecting different sections like the Library Page with the main database, enabling seamless data storage and retrieval.
- I introduced my team to version control and produced the first version of the project, **Lyfeline v1**, ensuring smooth collaboration and updates.
- I demonstrated how to access the database, create new tables in the Lyfeline_db, and manage data effectively, ensuring team-wide proficiency.

4.2.2 Visual Design and Implementation

I took charge of designing and building the **Library Page**, focusing on key elements:

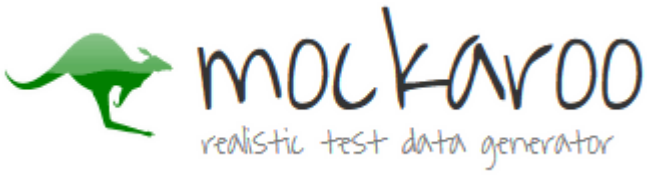
1. **Navigation Bar**:
    - Designed a responsive top bar featuring:
        - A hamburger menu for expanding navigation options.
        - The site name 'Lifelyne' prominently displayed at the center.
        - Help and settings buttons positioned on the right.
    - Added a 'Study Tracker' tab to the navigation bar, ensuring seamless integration with other sections.
2. **Search Area**:
    - Developed a horizontally aligned search bar, ensuring it was visually balanced with padding on both sides.
    - Styled and incorporated a functional search button to enhance usability.
3. **Book Display Section**:
    - Organized books in an alphabetical layout for easy browsing.
    - Added dropdown menus to each book, enabling users to mark their status as 'Read' (green), 'Reading' (yellow), or 'Awaiting'.
    - Implemented a 'Save' button for every book to allow users to bookmark items for future reference.

4.2.3 Database Design

I designed and implemented the **books table** schema in PostgreSQL to manage book data effectively. The schema was structured as follows:

```
CREATE TABLE books (
  id SERIAL PRIMARY KEY,
  title VARCHAR(255) NOT NULL,
  author VARCHAR(255) NOT NULL,
  status VARCHAR(50) DEFAULT 'Awaiting'
);
```

This table enabled efficient storage, retrieval, and management of book information, meeting the functional requirements of the Library Page. For sample data, I utilized **Mockaroo** to populate the table with realistic book entries, enhancing the development and testing process.



4.2.4 Enhancing Interactivity with JavaScript

To bring the Library Page to life, I wrote JavaScript code that:

- Allowed users to dynamically update book statuses through dropdown selections.
- Made the 'Save' button functional by storing user preferences in the database.
- Added responsiveness and interactivity to the navigation menu, improving the overall user experience.

4.2.5 Collaborative Efforts

- I proofread reports and provided constructive feedback to my peers, ensuring the quality of our documentation and screencast presentations.
- I ensured all CSS styles across the application were aligned for a seamless transition between pages, emphasizing consistency in design.

4.2.6 Versioning and Deployment

- I introduced versioning to the team, a practice that improved our workflow by keeping track of changes and enabling smooth rollbacks when needed. By creating **Lyfeline v1**, I established the foundation for future iterations.
- During deployment, I assisted in setting up the environment to ensure the backend and database were accessible and optimized for testing.

4.3 Challenges and Unused Contributions

Some of my ideas and initial work didn't make it to the final application due to time constraints or shifting priorities:

1. **University-Specific Library System**:
    - I envisioned a system where users could access a curated library based on their university. This feature would have required users to obtain a passkey (e.g., through a subscription) to access specific books provided by professors or institutions. Despite extensive research, implementing this feature required more time and collaboration with real-world universities, which was not feasible within the project timeframe.
2. **Advanced Filtering for Books**:
    - I worked on filters that would let users sort books by criteria such as genre, publication year, or author. However, this feature was set aside to focus on higher-priority tasks.
3. **Offline Functionality**:

- I explored methods to enable offline access to the Library Page. While promising, this feature was ultimately shelved as it exceeded the project's immediate scope.

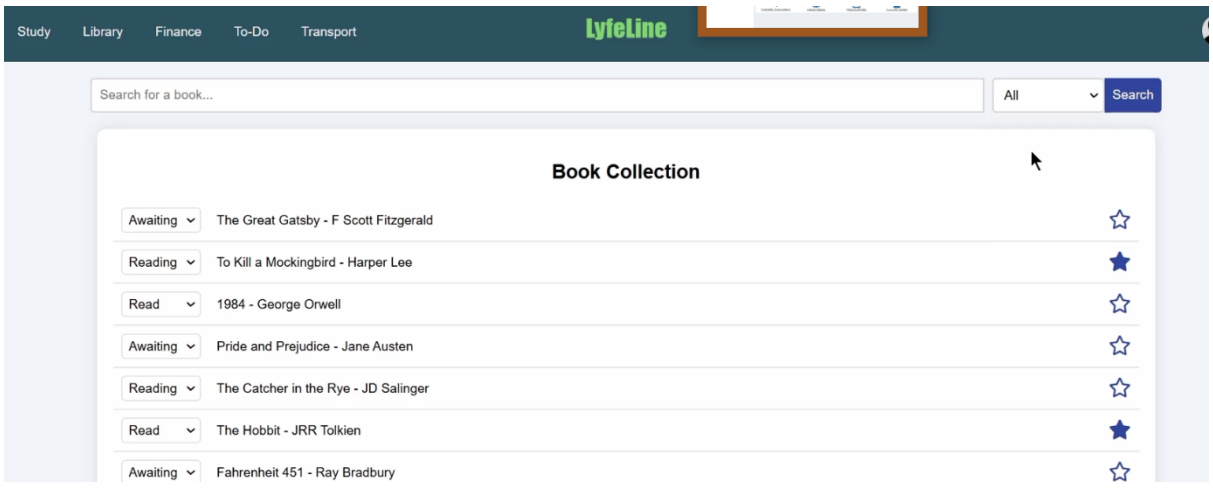
4. **Proof-of-Concept Features**:
   - I developed mockups for integrating a feature where students could upload their own reading materials or notes. Although this aligned with the goal of creating a versatile tool for students, the idea was deferred for future development.

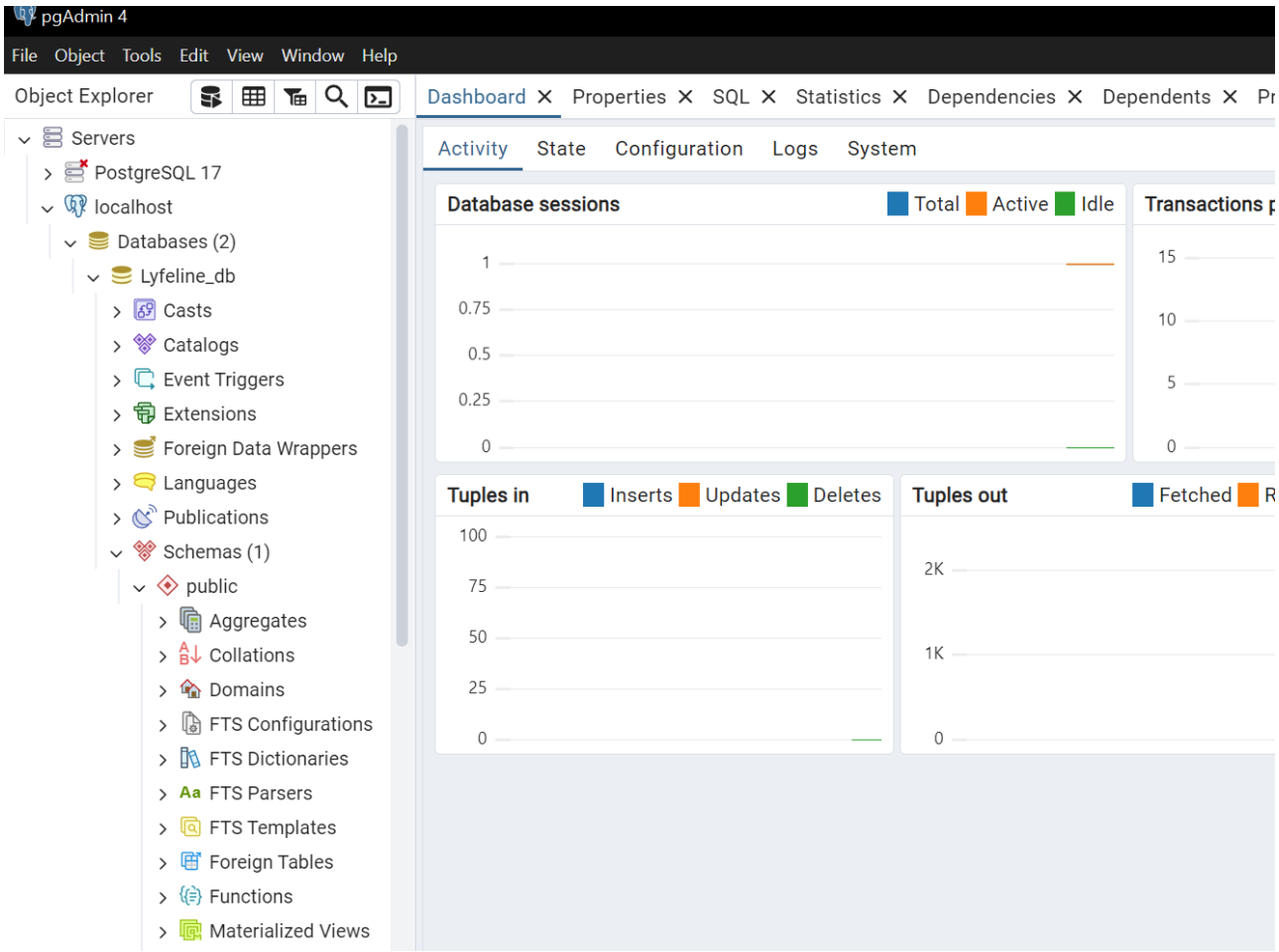
4.4 Reflection on Contributions

Working on the CS353 project provided an invaluable learning experience. I honed my skills in web development, database design, and user-centric design practices. Collaborating with my team taught me the importance of adaptability, effective communication, and finding a balance between ambitious features and project deadlines.

Although not all my contributions were included in the final product, the research and development process deepened my understanding of software development and enriched my problem-solving abilities. I also gained hands-on experience with tools like PostgreSQL, Node.js, and Mockaroo, which significantly boosted my technical expertise.

This project reinforced my interest in creating meaningful applications that address real-world problems. I hope to continue working on this project to bring ideas like the university-specific library system to life, aiming to support students and young adults like myself. Overall, my role in this project solidified my technical foundation and prepared me for future team-based endeavors in web development and beyond.




pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Servers
PostgreSQL 17
localhost
Databases (2)
Lyfeline_db
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Activity
State
Configuration
Logs
System
Database sessions
Total
Active
Idle
1
0.75
0.5
0.25
0
Transactions
15
10
5
0
Tuples in
Inserts
Updates
Deletes
100
75
50
25
0
Tuples out
Fetched
2K
1K
0

Chapter 5: Summary

Looking back on the 'LyfeLine' project, I can see just how much I've grown both as a developer and as a problem solver. The journey wasn't without its challenges, but it was exactly these challenges that pushed me to refine my skills and adapt to new situations. Building a library management system involved a wide range of tasks, from front-end design to back-end integration, and in the process, I learned a lot about balancing both the technical and creative sides of development.

Initially, I thought the front-end work would be straightforward. My goal was to create a clean and functional design for the library page, focusing on aesthetics and usability. But soon, I realized how much more complicated things could get. The project required me to think not only about how the site would look, but also how users would interact with it. For instance, designing the navigation bar with a hamburger menu, which at first seemed simple, turned out to be more tricky than I expected. I had to ensure it was responsive and worked well on different screen sizes, which took more time than I initially planned. This made me appreciate the importance of user-centered design, which ensures that the website is both functional and easy to navigate, especially when considering a range of user devices.

On the back-end, working with PostgreSQL was an eye-opener. I'd learned about database management in theory, but applying it to a real project highlighted just how crucial it is to ensure the system can handle data efficiently. Defining the database schema and ensuring everything was well-organized took more time than I expected, but the process gave me a much deeper understanding of the importance of good database design. There were moments when I felt like I was running in circles, especially when I had to tweak queries for performance optimization, but eventually, everything started to come together. Seeing the data flow smoothly from the database to the front-end after hours of coding felt incredibly rewarding.

A particularly memorable part of this project was when I implemented the dynamic book status dropdown and the 'Save for Later' functionality. At first, I thought these features would be simple, but they required a more careful approach. Not only did I have to make sure the buttons were easy to use, but I also needed to ensure that they worked well with the database. Implementing these features was a great exercise in learning how front-end and back-end systems need to communicate with each other seamlessly. I had to adjust the logic multiple times and even looked into AJAX for real-time updates, which was new to me. By the end of the project, I felt much more comfortable handling asynchronous requests and dynamic UI updates, both of which are essential skills for modern web development.

Another key lesson I took away from this project was the importance of testing and debugging. At first, I was eager to get everything running and see the final product, but I quickly learned that skipping over small tests could lead to bigger issues down the line. For example, sorting the books alphabetically seemed like a minor task, but I ran into problems when the sorting didn't work correctly or failed with special characters in book titles. These small bugs caused unnecessary frustration, but in hindsight, they taught me the importance of catching issues early, especially in something as fundamental as data display. As the project progressed, I made sure to test thoroughly, ensuring every feature was working before moving on to the next.

One unexpected challenge came when I had to integrate external libraries and resources. I hadn't anticipated how tricky it could be to make them fit into my project. For instance, when I added the hamburger menu, it clashed with some of my custom styles from a CSS library I was using. It wasn't a huge issue, but it required me to learn how to override default styles and apply my own without breaking the design. This was an important learning experience in working with third-party resources, as it highlighted the need to have a solid understanding of how these tools work and how to integrate them properly into a project.

Throughout the project, I also had to manage my time effectively. As I worked on various features and tackled unexpected issues, I quickly realized how important it was to stay organized. At first, I tried to tackle everything at once, but soon learned the value of breaking tasks down into smaller, more manageable chunks. I made a

habit of writing down my goals and deadlines, which helped keep me on track and ensured that I didn't get too overwhelmed. Managing time effectively, especially when you're working on multiple aspects of a project, is something I'll definitely carry forward into future work.

Adapting to new requirements was another lesson that this project really hammered home. The scope of the work shifted at times, and features that seemed minor at first turned out to require a lot more attention. For example, I originally didn't plan for a lot of customization on the user's side of things, but as I worked, I realized it would make the system more useful to allow users to save their progress. Flexibility was key, and it was important not to stick rigidly to a plan but instead to adapt and adjust as new needs arose. This experience showed me that flexibility is essential in software development, where the unexpected is almost always around the corner.

If I were to do this project again, I would focus more on accessibility. While I was concerned with the core functionality, I didn't put as much attention into making sure the site was accessible to all users, particularly those with disabilities. Small changes, like adding alt text to images or ensuring better keyboard navigation, would make a significant difference. I also think user feedback would've been helpful throughout the process. While I tested everything myself, it would have been great to involve others in the process to gain their perspective, especially in terms of user experience.

Overall, this project was a fantastic learning opportunity. It gave me the chance to apply everything I've learned in a real-world context, and although there were moments when I felt frustrated or stuck, overcoming those challenges made the project even more rewarding. It wasn't just about getting the website to work; it was about refining my problem-solving skills, learning to test and debug efficiently, and understanding the importance of adaptability in software development. As I look back, I can confidently say that I've gained valuable skills and insights that will benefit me in future projects and career opportunities.

Appendix

Minutes

Files

Minutes 09/10/2024

Attendance ● Joanna ● Alex ● Rory ● Presley ● Cian

Minutes of

- Minutes taken by Joanna this week.
- Members posted user stories and maps onto the teams channel.

● Miro wa sdecided to be used along with paper planning, link shared onto the teams channel. ● Discussed the name for the web application, yet to be decided.

● TFI api research for a feature to be integrated.

● Discussed user stories and decided on what to keep and what to discard.

● Discussed the most difficult part of the project, and amount of time to assign for each feature.

● Assigned in terms of difficulty, which part of the project to work on first.

● React.js agreed to be used for the user interface.

● Foundandagreed on which relational database to use (postgreSQL).

● Account creation and saving details was agreed to be the hardest part.

● Highpriority tasks: learn react.js, refresh SQL, start envisioning web application (Balsamiq)

● Start simple, build UI first, add more complex features later.

● Estimate size of each task: High priority tasks- 1 week, setting everything up- 21 week, building UI- 2 weeks. Minutes 16/10/2024

Minute taker: Cian Curran

Scrum Master: Rory Joseph O'Donoghue

Attendane

Joanna

Presley

Cian

Alex

Rory

Meer

Last Meeting

Revised SQL

Choose Story Map

Researched different additional features for

website Started Planning poker

Minutes of Meeting

Team has decided on website name, that being:

LifeLine. Set up Gitlab on nuim.ie.

Discussed work done from last meeting up until now.

Discussed UX/UI for website and planning on designing interface through wireframes.

Planning on work for the following week, start assigning roles.

Decided to stick with a web application for now instead of an application for the

phone. Had a Scrum Meeting going over all the work from the last meeting to now.

Roles

(Subject to change as project develops)

Frontend: Joanna, Presley, Rory, Alex, Tabitha

Backend: Cian, Meer

Work done Today

Alex: Working on prototype Wireframe for website

Group: Planned for everyone to design a wireframe to decide on the final UI for

website. Group: Discussed and created a timeline for future work.

Cian: Took minutes.

Joanna: Created start of timeline.

Group: Assigned wireframes to be designed by next week.

Group: Has decided to add in a 1 hour meeting for each week, MONDAY 1pm-2pm in person after CS310 Lab

TimeLine

16/10/2024 – 23/10/2024(Might push to 21st Monday after CS310 Lab): Each member of the Group will design a wireframe during this period.

23/10/2024 – over Midterm (video call over teams, time to be decided): Start Frontend code.

Work for Next Week

Group: Design a WireFrame for the website's UI

Joanna: Start working on a practice script for project description and presentation (optional).

Joanna: Wireframe's: Home Page, To-Do List

Presley: Wireframe's: Home Page, Library

Section Rory: Wireframe's: Home Page, Habit

Tracking Cian: Home Page, Login Page

Alex: Home Page, Finance Page

Meer: Home Page

Tabitha: Home Page

MONDAY 1pm-2pm in person after CS310 Lab

NOTE ON WIREFRAME'S

The design premise for the wireframes is SIMPLE, Alex has put an example Wireframe up on teams that can be used for inspiration. The hope is that all the wireframes from the group will be brought together, and the group will collectively decide on a design scheme based mostly on the home pages. After this we will be getting a start on the coding aspect of the project.

Minutes 23/10/2024

Week 5 Team Meeting – 23/10/2024

Minutes taken: Alex M
In Attendance: Alex, Rory, Joanna, Cian, Tabitha

ScrumMaster = Cian

Last week we discussed the foundations of our project and finalised the plan we wanted to take for the design of our application. We all decided that the idea we had was too broad for one person to decide on the design. To solve this we thought it would be good to each come up with a wireframe design, compare them to each other's in this week's stand up meeting and decide on what we like best about each persons and what we do not like.
This will help give us a much clearer understanding of what is expected of us in terms of the design.

To start the meeting, the 5 of us in attendance are going through our wireframes and presenting them to the group to explain our thought process and why we designed them the way we did. We all agreed that mine (Alex) was the one most in line with our overall vision. We mutually agreed that the log in page was going to be the first challenge to tackle in terms of actual coding. We discussed databases a little further, trying to figure out which database would be the best to use for the user sign in details.

We discussed timelines and planned a teams call for the middle of the mid term to discuss our code that we have done by now.

We decided that we could make a basic homepage that we could distribute that would let everyone work off of it so that all our code matches and is in line with our overall design goal.

Timeline:
By next Wednesday (30th of October) we will have the nav bar done and preferably would have started on each of our own pages that we were assigned last week. Then the following Monday after that, November 4th we will have our in person team meeting to show how well they pages link together and make any necessary changes that we may need.

End of the meeting:
For the end of the meeting we went over a few other details. We designed a basic navbar/general interface that we could all branch from to make our pages that we were assigned. Cian applied for the TFI Live API that would allow us showcase the live bus and train times on our project, Tabitha helped design a logo that we could use on our project, and Rory began work on the to do list by learning some basic react.

Work for next week:
Alex – Work on the finance page
Rory – Work on the to do list
Cian – Working on the implementation of the TFI API
Joanna – Working on the home / landing page
Tabitha – Working on the log in / sign up page
Meer and Presley – Provided some work on the teams as they were unable to attend, the minutes to be uploaded so they can catch up on what we went over in todays meeting

Minutes 13/11/2024

Attendance :Presley Joana Tabitha Meer Alex

- We held a scrum meeting regarding our projects
- We discussed the issues we encountered and how we are were going to fix them
- We set up an online meeting for Friday 15th
- We discussed the upcoming deadlines e.g report
- Joana volunteered to put the entire project together

- Tabitha fixed her nav bar as it wasn't matching the

UI Minutes 20/11/2024

Group project (team 17) minutes week 20/11/2024

Attendees: Alex, Rory, Joanna, Tabitha Cian.

Presley at interview.

Went over progress and looking at tasks that need to be done over next few weeks.

Discussing next few weeks for project. How we are going to integrate all databases together in backend. - 10mins

Minutes: Rory

SCRUM Master: Alex

SCRUM Meeting:30 mins

Went over work for previous weeks

Rory got backend up for Todo list

Alex got backend up for finance

page Tabitha got the login page up.

Pieced all front end up together, had difficulty integrating Presley's code. Issue is hoped to be resolved on Friday.

Future focus:

Testing phase->ironing out any minor errors and debugging.

Uploading code to GITHUB to have our code more

collaborative. Everyone will do some research on GIT before

meeting on Friday. Planned a meeting for Friday 22nd@3pm.

Last 30 mins were spent setting github up.

Minutes 22/11/2024

22/11/2024 Minutes

Attendees: Tabitha, Alex, Meer, Presley, Rory, Joanna, Cian

Length: 20 mins

Topics Covered:

Recapped what everyone is working on,

Everyone joined the GitLab

More files were added.

Discussed issues and aspects that were not working

Planned to test we are going to do next Wednesday.

Minutes 04/12/2024

Minutes for 04/12/2024

SCRUM Master: Cian
Attendance: 7/7

-Discussed the layout of the report
-Agreed to do the individual part of report first (rough draft due for Friday Meeting 06/12/24)
-Plan to work on team part of report for next Wednesday lab
-Updates on bus schedule part of our project (work in progress, ~75% done)
-Full individual part for report hopefully by next Wednesday Lab
-Plan to do screencast, next Wednesday (book room in library yet to be done)
-More meetings planned for this week (Friday, Saturday, following Monday)

Minutes 11/12/2024

Minutes for 11/12/2024 (At the Moment)

SCRUM Leader: Alex

Attendance: Joanna, Cian, Alex, Rory, Presley

-Discussed the finalisation of the project

-Pushed remaining code and folders onto the GitLab

-Reminded team mates to push code/debug code

-Distributed group work for following week

Minutes 16/12/2024

Other User Stories and Story Maps

User Stories

1. First Year Student (Beginner user): Dave has joined Maynooth University after taking a year out after the leaving cert and is now struggling to balance study time with social activities. After finding out about Lifestyle (Website), Dave configures the study reminder feature choosing to receive study notifications based on free time detected in his online timetable. Dave later on gets a notification during an hour-long gap between lectures to start studying. Dave clicks the "Start Studying" button, which mutes notifications and enables focus mode. He also decided to enable background music to play soft instrumentals. After studying for an hour, Dave earns a streak. The next day, the app sends another reminder, helping Dave build a consistent study routine.

2. Final Year Student: Kate has finished her final year project and is now waiting for her graduation day. Kate starts to get concerned about trying to save money after collage. She logs into Lifestyle and inputs her weekly earnings from a part-time job and sets savings goals to help organise her future purchases. The website tracks her progress towards savings goals and suggests tips, like cutting unnecessary subscriptions or spending less on coffee. It also reminds her when upcoming bills are due. Kate starts to feel more financially prepared for post-grad life.

3. International Student: Adnan, an international student, is trying to find the best way to get back to his apartment after late-night classes. He is still unfamiliar with the campus and the layout outside of it. Adnan uses the bus tracker feature in the website to search for buses departing from the closest public bus stop to his current location. The website shows real-time bus schedules and highlights the buses that can take Adnan directly to his destination. Adnan selects the most convenient route, gets real-time updates on bus delays, and sets an alert to remind him 5 minutes before the bus arrives at the stop. Adnan successfully navigates public transportation with ease, arriving home on time without the stress of figuring out the route manually.

User Stories:

Types of users: highschool student?, young adult, college student, graduate?

Want to: login/register(3), look over my balance, look into my expenses, set budgets, check bus times, set an alarm/reminder, search routes, add to my streaks, look through my [streak related] stats.

User Management:

As a student I want to register on the website so that I can access what it's providing.

As a student I want to login into the website so that I can access what it's providing to me specifically.

As a student I want to logout of the website so that I can keep my information secure.

Finances:

As a student I want to look over my balance so that I know how much money I can spend.

As a student I want to look into my expenses so I can set budgets.

As a student I want to set budgets so that I can be more responsible with my money.

Commute:

As a student I want to check the different bus/train times so that I can be ready on time.

As a student I want to set an alarm/reminder so that I won't forget or miss the time.

As a student I want to search different routes so that I can find the most efficient or shortest route.

Streaks:

As a student I want to add to my streaks so that I can be consistent.

As a student I want to look through my [streak related] stats so that I can know how far I have come.

Image links

Burndown chart-> Burndown Chart: What Is It & How to Use one for Agile?

What is a Story map-> Story Maps | Reading Rockets

Continuous Testing-> Agile Testing: Definition, Benefits & Disadvantages

Sample Minutes File-> Free Business Meeting Minutes Template - PDF | Word – eForms

Sprint Graphic-> What Are Sprints? - Hygger.io Guides

Planning poker->planning poker cards - Search Images
Mockaroo - > Mockaroo logo
Insights -> Insights display

Research Links

The SCRUM Process-> What is Scrum? | Scrum.org

Learning React-> React in 100 Seconds

Recapping MySQL-> MySQL - The Basics // Learn SQL in 23 Easy Steps

Javascript, HTML and CSS recap-> HTML, CSS, and Javascript in 30 minutes

Article on Continuous Testing-> Continuous Testing in Software Testing -

GeeksforGeeks What is a user story-> User Stories | Examples and Template |

Atlassian

Node js - > Connecting pages

Pgadmin -> Database implementation

- Discussed user stories and decided on what to keep and what to discard.
- Discussed the most difficult part of the project, and amount of time to assign for each feature.
- Assigned in terms of difficulty, which part of the project to work on first.
- React.js agreed to be used for the user interface.
- Foundandagreed on which relational database to use (postgreSQL).
- Account creation and saving details was agreed to be the hardest part.
- Highpriority tasks: learn react.js, refresh SQL, start envisioning web application (Balsamiq)
- Start simple, build UI first, add more complex features later.
- Estimate size of each task: High priority tasks- 1 week, setting everything up- 21 week, building UI- 2 weeks. Minutes 16/10/2024

Minute taker: Cian Curran

Scrum Master: Rory Joseph O'Donoghue

Attendanc

e Joanna

Presley

Cian

Alex

Rory

Meer

Last Meeting

Revised SQL

Choose Story Map

Researched different additional features for

website Started Planning poker

Minutes of Meeting

Team has decided on website name, that being:

LifeLine. Set up Gitlab on nuim.ie.

Discussed work done from last meeting up until now.

Discussed UX/UI for website and planning on designing interface through wireframes.

Planning on work for the following week, start assigning roles.

Decided to stick with a web application for now instead of an application for the

phone. Had a Scrum Meeting going over all the work from the last meeting to now.

Roles

(Subject to change as project develops)

Frontend: Joanna, Presley, Rory, Alex, Tabitha

Backend: Cian, Meer

Work done Today

Alex: Working on prototype Wireframe for website

Group: Planned for everyone to design a wireframe to decide on the final UI for website.

Group: Discussed and created a timeline for future work.

Cian: Took minutes.

Joanna: Created start of timeline.

Group: Assigned wireframes to be designed by next week.

Group: Has decided to add in a 1 hour meeting for each week, MONDAY 1pm-2pm in person after CS310 Lab

TimeLine

16/10/2024 – 23/10/2024(Might push to 21st Monday after CS310 Lab): Each member of the Group will design a wireframe during this period.

23/10/2024 – over Midterm (video call over teams, time to be decided): Start Frontend code.

Work for Next Week

Group: Design a WireFrame for the website's UI

Joanna: Start working on a practice script for project description and presentation (optional).

Joanna: Wireframe's: Home Page, To-Do List

Presley: Wireframe's: Home Page, Library Section

Rory: Wireframe's: Home Page, Habit Tracking

Cian: Home Page, Login Page

Alex: Home Page, Finance Page

Meer: Home Page

Tabitha: Home Page

MONDAY 1pm-2pm in person after CS310 Lab

NOTE ON WIREFRAME'S

The design premise for the wireframes is SIMPLE, Alex has put an example Wireframe up on teams that can be used for inspiration. The hope is that all the wireframes from the group will be brought together, and the group will collectively decide on a design scheme based mostly on the home pages. After this we will be getting a start on the coding aspect of the project.

Minutes 23/10/2024

Week 5 Team Meeting – 23/10/2024

Minutes taken: Alex M
In Attendance: Alex, Rory, Joanna, Cian, Tabitha

ScrumMaster = Cian

Last week we discussed the foundations of our project and finalised the plan we wanted to take for the design of our application. We all decided that the idea we had was too broad for one person to decide on the design. To solve this we thought it would be good to each come up with a wireframe design, compare them to each other's in this week's stand up meeting and decide on what we like best about each persons and what we do not like.
This will help give us a much clearer understanding of what is expected of us in terms of the design.

To start the meeting, the 5 of us in attendance are going through our wireframes and presenting them to the group to explain our thought process and why we designed them the way we did. We all agreed that mine (Alex) was the one most in line with our overall vision. We mutually agreed that the log in page was going to be the first challenge to tackle in terms of actual coding. We discussed databases a little further, trying to figure out which database would be the best to use for the user sign in details.

We discussed timelines and planned a teams call for the middle of the mid term to discuss our code that we have done by now.

We decided that we could make a basic homepage that we could distribute that would let everyone work off of it so that all our code matches and is in line with our overall design goal.

Timeline:
By next Wednesday (30th of October) we will have the nav bar done and preferably would have started on each of our own pages that we were assigned last week. Then the following Monday after that, November 4th we will have our in person team meeting to show how well they pages link together and make any necessary changes that we may need.

End of the meeting:
For the end of the meeting we went over a few other details. We designed a basic navbar/general interface that we could all branch from to make our pages that we were assigned. Cian applied for the TFI Live API that would allow us showcase the live bus and train times on our project, Tabitha helped design a logo that we could use on our project, and Rory began work on the to do list by learning some basic react.

Work for next week:
Alex – Work on the finance page
Rory – Work on the to do list
Cian – Working on the implementation of the TFI API
Joanna – Working on the home / landing page
Tabitha – Working on the log in / sign up page
Meer and Presley – Provided some work on the teams as they were unable to attend, the minutes to be uploaded so they can catch up on what we went over in todays meeting

Minutes 13/11/2024

Attendance :Presley Joana Tabitha Meer Alex

- We held a scrum meeting regarding our projects
- We discussed the issues we encountered and how we are were going to fix them
- We set up an online meeting for Friday 15th

- We discussed the upcoming deadlines e.g report
- Joana volunteered to put the entire project together

- Tabitha fixed her nav bar as it wasn't matching the

UI Minutes 20/11/2024

Group project (team 17) minutes week 20/11/2024

Attendees: Alex, Rory, Joanna, Tabitha Cian.

Presley at interview.

Went over progress and looking at tasks that need to be done over next few weeks.

Discussing next few weeks for project. How we are going to integrate all databases together in backend. - 10mins

Minutes: Rory

SCRUM Master: Alex

SCRUM Meeting:30 mins

Went over work for previous weeks

Rory got backend up for Todo list

Alex got backend up for finance

page Tabitha got the login page up.

Pieced all front end up together, had difficulty integrating Presley's code. Issue is hoped to be resolved on Friday.

Future focus:

Testing phase->ironing out any minor errors and debugging.

Uploading code to GITHUB to have our code more

collaborative. Everyone will do some research on GIT before

meeting on Friday. Planned a meeting for Friday 22nd@3pm.

Last 30 mins were spent setting github up.

Minutes 22/11/2024

22/11/2024 Minutes

Attendees: Tabitha, Alex, Meer, Presley, Rory, Joanna, Cian

Length: 20 mins

Topics Covered:

Recapped what everyone is working on,

Everyone joined the GitLab

More files were added.

Discussed issues and aspects that were not working

Planned to test we are going to do next Wednesday.

Minutes 04/12/2024

Minutes for 04/12/2024

SCRUM Master: Cian
Attendance: 7/7

-Discussed the layout of the report
-Agreed to do the individual part of report first (rough draft due for Friday Meeting 06/12/24)
-Plan to work on team part of report for next Wednesday lab
-Updates on bus schedule part of our project (work in progress, ~75% done)
-Full individual part for report hopefully by next Wednesday Lab
-Plan to do screencast, next Wednesday (book room in library yet to be done)
-More meetings planned for this week (Friday, Saturday, following Monday)

Minutes 11/12/2024

Minutes for 11/12/2024 (At the Moment)

SCRUM Leader: Alex

Attendance: Joanna, Cian, Alex, Rory, Presley

-Discussed the finalisation of the project

-Pushed remaining code and folders onto the GitLab

-Reminded team mates to push code/debug code

-Distributed group work for following week

Minutes 16/12/2024

Other User Stories and Story Maps

User Stories

1. First Year Student (Beginner user): Dave has joined Maynooth University after taking a year out after the leaving cert and is now struggling to balance study time with social activities. After finding out about Lifestyle (Website), Dave configures the study reminder feature choosing to receive study notifications based on free time detected in his online timetable. Dave later on gets a notification during an hour-long gap between lectures to start studying. Dave clicks the "Start Studying" button, which mutes notifications and enables focus mode. He also decided to enable background music to play soft instrumentals. After studying for an hour, Dave earns a streak. The next day, the app sends another reminder, helping Dave build a consistent study routine.

2. Final Year Student: Kate has finished her final year project and is now waiting for her graduation day. Kate starts to get concerned about trying to save money after collage. She logs into Lifestyle and inputs her weekly earnings from a part-time job and sets savings goals to help organise her future purchases. The website tracks her progress towards savings goals and suggests tips, like cutting unnecessary subscriptions or spending less on coffee. It also reminds her when upcoming bills are due. Kate starts to feel more financially prepared for post-grad life.

3. International Student: Adnan, an international student, is trying to find the best way to get back to his apartment after late-night classes. He is still unfamiliar with the campus and the layout outside of it. Adnan uses the bus tracker feature in the website to search for buses departing from the closest public bus stop to his current location. The website shows real-time bus schedules and highlights the buses that can take Adnan directly to his destination. Adnan selects the most convenient route, gets real-time updates on bus delays, and sets an alert to remind him 5 minutes before the bus arrives at the stop. Adnan successfully navigates public transportation with ease, arriving home on time without the stress of figuring out the route manually.

User Stories:

Types of users: highschool student?, young adult, college student, graduate?

Want to: login/register(3), look over my balance, look into my expenses, set budgets, check bus times, set an alarm/reminder, search routes, add to my streaks, look through my [streak related] stats.

User Management:

As a student I want to register on the website so that I can access what it's providing.

As a student I want to login into the website so that I can access what it's providing to me specifically.

As a student I want to logout of the website so that I can keep my information secure.

Finances:

As a student I want to look over my balance so that I know how much money I can spend.

As a student I want to look into my expenses so I can set budgets.

As a student I want to set budgets so that I can be more responsible with my money.

Commute:

As a student I want to check the different bus/train times so that I can be ready on time.

As a student I want to set an alarm/reminder so that I won't forget or miss the time.

As a student I want to search different routes so that I can find the most efficient or shortest route.

Streaks:

As a student I want to add to my streaks so that I can be consistent.

As a student I want to look through my [streak related] stats so that I can know how far I have come.

Image links

Burndown chart-> Burndown Chart: What Is It & How to Use one for Agile?

What is a Story map-> Story Maps | Reading Rockets

Continuous Testing-> Agile Testing: Definition, Benefits & Disadvantages

Sample Minutes File-> Free Business Meeting Minutes Template - PDF | Word – eForms Sprint Graphic-> What Are Sprints? - Hygger.io Guides

Planning poker->planning poker cards - Search Images

Research Links

The SCRUM Process-> What is Scrum? | Scrum.org Learning React-> React in 100 Seconds

Recapping MySQL-> MySQL - The Basics // Learn SQL in 23 Easy Steps Javascript, HTML and CSS recap-> HTML, CSS, and Javascript in 30 minutes

Article on Continuous Testing-> Continuous Testing in Software Testing - GeeksforGeeks What is a user story-> User Stories | Examples and Template | Atlassian