ScheduleSphere

"Managing Schedules, Building Bridges"

Course: CSC 392

BSc Information Science

Group members

Martin Nhemachena	**202102839**
Patience Chemane	**202102447**
Bonginkosi Mkhabela	**202103227**
Minenhle Mangwe	**202102938**
Thembeka Nzuza	**202003104**

Table of contents

Introduction

ScheduleSphere is an information delivery android app designed using Kotlin DS, Android Room and persistence. It provides a dynamic timetable that enables students to add, delete and edit their courses giving them a tool to manage their schedule throughout the week. This application also provides information on the memos, activities and announcements within the school, keeping both the students and staff aware of current events. This app aims to centralise information delivery within the school, and enable both staff and students to use one platform for information delivery.

Objectives

1. **Efficient Time Management**: The goal is to help students with time management and the planning of schedules.
2. **Seamless Communication**: The memos and announcements mean that students can easily stay updated on any information from the palm of their hands.

Key Features

- **Timetable Management**: Students can add and delete courses on their dynamic timetable enabling them to effectively plan out their days.
- **Memos and Announcements**: School administrators and lecturers can send out memos and faculty announcements to those who are concerned.
- **School Activities Calendar**: This application will provide a calendar of what is happening within the school, including sporting events, seminars and other types of events.

Assumptions and constraints

- Team members will attend all meetings
- Team members will meet deadlines
- Team members will work on the project outside of class to finish it on time
- The application is being designed to only run on Android devices

Risks and Mitigation

The risks of this particular project is that data integrity of the users might be compromised, furthermore users might just find it too difficult to understand or not understand what the application is trying to accomplish. Each team member will be kept up to date, and inform managing team of any changes or difficulties that might affect their ability to complete their tasks on time. Lastly a large array of technical issues can arise including improper implementation of the android studio environment. The best way to mitigate the risks would be extensive testing and proper implementation of security features and a proper implementation of the software engineering process to ensure effective results.

Client/User

The StudySphere application has various stakeholders, different communities within the school. The students would be using the application for managing their timetables then the lecturers can use it to post announcements whilst the school administration could use this application to post memos and other announcements.

Users

Students-The students would be using this application for managing the time table and receiving information about the memos and the activities within the school.

Lecturers**-**Lecturers would be able to post announcements about any developments with the courses they instruct which facilitates and enhances the learning process. Currently most of these announcements are conveyed through word of mouth or whatsapp messages which leaves a portion of the concerned parties unaware of certain information.

School Administrators-School administration would be able to post memos to the application which helps convey information on a centralised platform that will enable to share information with all concerned parties and serve as a supplement to the already available information delivery system which is the email.

Client

In the context of the University of Eswatini, the institution would be the client of the application. The university is responsible for facilitating tools to enhance communication and learning. This means that the institution would be the one determining the features that are needed to help them deliver information more effectively.

Organisation

The project has four personnel's working on it, the project manager, architect, programmer, designer and tester. The project manager is responsible for managing the team and keeping track of progress and performance of project, along the performance and efficiency of team members. The manager has the role to recruit and delegate tasks to team members making sure each task is completed in time.

The architect will work hand in hand with the project manager and ensure that communication and networking is in place. They will also work with the programmer when developing the software system of the project.

The programmer will assume the role of choosing which language of code to use for our calendar. Writing and testing code functionality will be their main role, while also identifying and correcting coding errors. Preventing security breaches and data links.

Standards, guidelines and procedures

- Stand-up meetings- there will be regular short meetings the team will attend. This meetings will help team members to share what they are working in and the challenges they are facing at any particular stage of the development process. The meetings will also help in keeping everyone on track and prevent problems from developing.

- Code formats and style guides- the guides will specify how to format code, indentation, spacing, naming conventions and commenting. This will make code easier to read this reduces errors.

- Static code analysis- this are tools that will assist the programmer in identifying code errors or vulnerabilities without actually running the code. This will help catch bugs early to make the code more secure.

- System testing- this is a must for this project. System testing tests the entire software system from the users' perspective to see if our application will be user friendly. This will be done before the application is released.

- Security reviews- this involves systematically assessing the code to check for vulnerabilities. Our team will do this manually and with the aid of automated tools.

Resources

To oversee the development process, coordinate tasks and ensure the project stays on schedule, we are going to need a project manager. This person will maintain the agreed project plan while also applying the risk management processes to try and bridge communication as mentioned above. A manager also provides leadership in trying to implement our project. They will also be responsible for managing our budget which is very important if we want this application to work and be beneficial to its users.

To ensure the success of this app, our project manager will be qualified. They are basically responsible for the success of every project as they maintain project files which have reports, invoices and payment records to every plan and design.

Using a UI/UX designer will also ensure that our application is not only visually appealing but also user- friendly to students. This is another resource that we are going to use because it allows colour schemes and branding which will make it easy for lecturers and mostly students, to read and understand. Colour is also tied to feelings and the UI/UX will allow us to select fonts that will automatically enhance readability.

To store and manage data like all the schedules, time tables and records, we are going to need a database server. It will be used to store and manage databases that will be stored on the server so that it will provide access for the authorised users which are the students, lecturers and the administration. This database server will make sure that it keeps all our data in a central location that will make it easy to back it up regularly and also allow users to access it across the network.

Project phases

Our team will use the Agile model because it will help ScheduleSphere adapt quickly to change requests and help us complete the project faster. It emphasizes having a team of efficient members and enhancing communication among them. This model is beneficial because it facilitates for frequent modification, flexible project schedules and budgets and it puts the customer at the center of the development process, ensuring that requirements are met.

Project phases are simply the building blocks that we will use to bring ScheduleSphere to life.

1. Planning and analysis

This entails assessing the viability of our project by analysinganalysing technical feasibility, target market and the required resources. Moreover the user/ client's needs are taken into consideration to assist in defining the features and functionality of the software. Our project manager ensured that the team understands what the University wants from this application. The team made use of interviews and observations to get a clear understanding of the current system in place. This phase is important because it provides a clear picture of the project, which further helps us estimate finances, human resource and timeframe to complete the project.

2. Design

The design phase has to do withto do with curating how users will interact with ScheduleSphere, our focal point being usability, accessibility, completeness and aesthetics. It involves defining all the specifications of the application, structure and components of the application. This includes data structures, algorithms, APIs, servers, just to name a few.

3. Implementation

This phase has to do with putting our design plan into action. Our team has a programmer who will write code for each algorithm using the suitable programming language. This code is the one that basically implements the application's functionality. During this phase, unit testing will be introduced to ensure that each unit individual of code works and further testing how our various components work together. This means our team tester and programmer will be working closely together during this phase.

4. Testing

This has to do with ensuring or verifying whether ScheduleSphere meets the specified requirements. Our team comprisescomprises a tester who will assess the performance, security, and other non-functional features of the application. The tester will also perform acceptance testing which has to do with testing the application with actual users to check its usability and if it meets their expectations.

5. Deployment and maintenance

At this stage, ScheduleSphere will be ready and available for use on the androidon the android system. The application will be presented to the University. All testing would have been donebeen done and errors fixed. During the maintenance phase, new features may be added and also improveimprove the application's performance and scalability.

6. Monitoring and support

In this phase, the performance of StudySphere is monitored to identify any possible issues. It involves the provision of support to students, lecturers and staff who are experiencing any issues with the application. Moreover, it has to do with ensuring that the software is up to date with the latest security patches and updates.

Task	**Duration (days)**	**Start date**	**End date**	**Person responsible**	**Predecessor**
A- Planning	5	4 Dec	9 Dec	Project manager	
B- Design	7	11 Dec	19 Dec	Designer, architect	A
C- Implementation	25	20 Dec	16 Jan	Programmer	A,B
D- Testing	5	18 Jan	23 Jan	Tester/quality analyst	C
E- Deployment	2	24 Jan	26 Jan		A,B,C,D

Implementation

The resources and tools needed to support implementation of a time management software/application will vary depending on the complexity of your project, your budget, and your team's technical skills. However, for Schedulesphere, these are the essential categories we considered:

Development Resources:

Developers: we have a skilled developer to build the software itself.

Project Manager: A project manager oversees the development process, ensuring tasks are completed on time and within budget.

UI/UX Designer: our skilled designer will create the user interface and user experience, ensuring the software is intuitive, visually appealing, and easy to use.

QA Testers: Quality assurance testers will identify and report bugs and ensure the software is functioning correctly before launch.

Development Tools:

Programming Languages and Frameworks: for Schedulesphere we made use of Kotlin DS. Kotlin is concise, readable, and easy to learn.Static typing and null safety help create reliable, maintainable code that is easy to troubleshoot.Being a JVM language, Kotlin gives you great performance and an ability to leverage an entire ecosystem of tried and true Java libraries.

Version Control Systems: this software application made use of Git help manage code versions, track changes, and enable collaboration among developers.

Integrated Development Environments (IDEs): IDEs provide a platform for writing, testing, and debugging code, with features like syntax highlighting and code completion. This software application made use of Android Studio.

Test Management Tools: Platforms like Jira, TestRail, or Azure DevOps help manage test cases, track progress, and report defects.

Performance Testing Tools: Tools like JMeter or LoadRunner will help simulate real-world load and identify performance bottlenecks.

Bug Tracking Systems: Tools like Asana or Trello will help track and manage bugs and issues identified during development and testing.

Additional Resources:

Hosting: Room, a database in android studio will be used to store data like the activities, memos, timetables etc.

Cloud Storage: to store users' information or data, we used android account manager.

Analytics Tools: we intend to integrate analytics tools like Google Analytics to track the behaviour and identify areas of improvement for the software application.

Marketing and User Acquisition: to market our and reach out audience we use social media like Facebook since it is most popular around campus.

Quality Assurance

Ensuring the quality of ScheduleSphere is crucial, as it directly impacts user productivity and satisfaction. Our team will constantly perform a quality check on the software regularly and the client will be present to ensure that the project meets the required standards. Here are some key aspects we took into consideration for quality assurance (QA):

Functional Testing:

Core Features: We aim to Test the core functionalities like task creation, scheduling, time tracking, calendar integration, reminders, and notifications. Ensure everything works as intended and caters to various time management styles.

Reporting and Analytics: We intend Verify the accuracy and clarity of reports for time spent, task completion, and productivity trends. Ensure data is visualized effectively and actionable insights are provided.

Accessibility: We will also ensure the software is accessible to users with disabilities, following WCAG guidelines for keyboard navigation, screen reader compatibility, etc.

Platform Compatibility: we aim to to test the software across different operating systems (desktop, mobile), browsers, and devices to ensure consistent performance and usability.

Non-Functional Testing:

Performance: we intend to test the software under various load conditions (simultaneous users, large data) to ensure responsiveness and stability.

Security: we aim to implement encryption and security measures to protect user data and prevent unauthorized access. Conduct penetration testing to identify vulnerabilities.

Usability: We aim to conduct usability testing with real users to assess the software's intuitiveness, learnability, and overall user experience. Gather feedback and iterate on the design based on user input.

Additional Considerations:

Documentation: Ensure clear and comprehensive user documentation is available for all features and functionalities. We are using Google Doc for this purpose.

Error Handling: for our error handling, we will test how the software handles errors and unexpected situations, to provide informative messages and guide users towards solutions.

Regression Testing: After fixing bugs or implementing new features, we intend to re-test previously working functionalities to ensure nothing regressed and broke during the changes.

Changes

There are various potential changes that can be made to ScheduleSphere to improve its effectiveness and functionalities, taking into consideration the users' needs ad challenges. Our developers can create advanced friendly user tools, adaptive interface, security and enhance task management.

Financial plan

There is no budget allocated to the ScheduleSphere team. However, we will make sure that the project is efficient enough.

Conclusion

ScheduleSphere is an android application that aim to centralize information dissemination within the institution. The key objectives of this project include improved communication and collaboration, enhanced task visibility, and time management. This application will unlock a path to better organization and focus to our users.