Unit

University Social Media

Development Plan | Version 1.2

January 23, 2020

Team: Adeel Asghar, Tyler Gross, Palak Patel, and Hala Ali

Clients: Thomas Anter and Jahnu Best

GTA: Son Dang

Revision Table

Date	Description	Authors	Comments
1-21-20	Version 1.0	Adeel Asghar Hala Ali Tyler Gross Palak Patel	First draft sent to GTA.
1-22-20	Version 1.1	Adeel Asghar Hala Ali Tyler Gross Palak Patel	Final Version
4-15-20	Version 1.2	Adeel Asghar	Added logo

Table of Contents

1.1 Project Overview

While there are many resources for university students, it is difficult for them to find and keep track of them. The Collaborative University Forum Android App aims to create a space for WSU (Wayne State University) students to collaborate with one another for help with studying for classes, and any other general question they can post on the forum. The features that will help meet these objectives include a forum and potentially, shared flash cards.

1.2 Project Purpose, Objectives, and Scope

The purpose of this project is to develop a collaborative university forum in the form of an android application. Our application will allow WSU students to easily communicate with each other as well as answer each other's questions. Students will be able to search for forums specific to classes they are taking or interested in. These forums can be used to ask questions as well as to sell and buy old class materials. The development team will be responsible for developing this app from its foundation which includes designing an easy to use UI as well as focusing on the back-end components.

The scope of this project is creating the student forum for WSU students. The forum contains a login page for users to access the application, a home page where users can view forums that they have subscribed to, and a search function so users can easily find the forums that they are interested in. After consulting with our clients, our stretch goals for the app include shared flashcards and Wayne State student verification through the wayne.edu search directory for people in order to prevent non-WSU students from accessing the app.

1.3 Team Organization

Team Lead: Adeel Asghar

The team lead is required to facilitate weekly meetings between the client and the rest of the team as well as lead additional team meetings for teamwork collaboration. The team lead is also required to delegate tasks to the team members and hold them accountable to the deadlines for the project. Another responsibility of the lead is to enforce problem resolution policies and oversees scheduling meetings with the GTA and possibly with the instructor based on the severity of the offenses committed by team member.

Documentation and Presentations Lead: Hala Ali

The documentation and presentations lead proofreads the content on these deliverables and ensures that there are no simple mistakes such as typos or other syntactical errors. Also, this lead reads over the content to determine that the content is thorough enough in explaining our developmental process for documentation. The lead reviews content for presentations to make

sure that there is consistency in the form of a parallel structured format and a clear, concise message of the progress that is made for the mobile app at each milestone.

Backend/Database Lead: Tyler Gross
The back-end/database lead is responsible for developing the backend of the product using Kotlin and connecting to Amazon Web Services whether this includes user authentication or database storage.

Frontend/UI Lead: Palak Patel
The front-end/UI lead is responsible for designing a convenient, easy to use UI that university students will be able to use effectively. The interface should provide the user with a happy UX by having navigability and consistent design.

1.4 Problem Resolution Policies

If the team member misses a weekly meeting with the client, the issue will be dealt internally with the remaining team members. The team members will communicate with the tech lead and the tech lead will be responsible for confronting the team member about missing the meeting. Documentation of team members missing a meeting will all be recorded through meeting minutes with the client. A second repeat offense will result in the team members and the tech lead reaching out to the GTA to discuss what action should be taken with the team member moving forward.

If the team member misses three or more meetings without any justifiable cause, the remaining team members will schedule a meeting with the GTA and the instructor, presenting a case of the team member being absent and showing unprofessional conduct. The punishment for such an offense shall be determined by the instructor at this point. If the team member did have an emergency that he/she could not attend a meeting with the client, that team member must show proof of the situation that led to them missing the meeting.

In the case that a team member feels that they may miss a deadline, they must let the rest of the group members know at least 48 hours before the deadline about whatever issues they are having, so those issues can be resolved as soon as possible. If they fail to notify the team, they will be trusted with less responsibility going forward. The first offense will be resolved internally by a meeting with the remaining team members and the tech lead. The tech lead shall confront the team member about their failure to meet the deadline and have the team member submit a detailed plan of work specifically to the tech lead and the GTA in the following weeks to ensure that the team member is motivated to meet all future deadlines.

A second offense will result in the tech lead providing documentation of the team member's plan of work he/she has submitted in a meeting with the GTA and how they have failed again to meet the deadline, despite the helpful intervention method from the tech lead to get that person on track. They will discuss together what needs to be done about this team member. If a team member misses three or more deadlines, all remaining team members will meet with the GTA and the instructor to present all documentation on that team member in the form of what they completed and what they failed to do. The punishment for such an offense shall be determined by the instructor at this point.

1.5 Project Plan

We will meet twice a week among ourselves on Wednesdays for a few hours after our client meeting and on Fridays from 5pm-7pm in the UGL (Undergraduate Library) on campus. We have decided to meet with our clients for one hour every Wednesday from 5:30pm to 6:30pm in the UGL.

Our iterations, or sprints, will be 1 weeklong. After our discussion with the clients, we have decided what our high-level deliverables will be throughout our developmental process. For the first prototype, the client requested for us to have a login and password UI created, a functioning forum that will allow a user to post a few sentences to one of the class feeds. Tags should be associated to the post and sub class feeds should also be implemented. A temporary cache like SQLite can be used to store user information if an AWS can't be implemented in time. Our second prototype should have the AWS server implemented, the login should be refined with a possible and forum should be refined to allow commenting. We will continue to keep contact with our clients and based on our progress, we will decide on what the deliverables will be for the final prototype.

1.6 Configuration Management Plan

All Version control will be handled in a private GitHub repository. Project task assignments will be provided by the team.

A Trello board has been shared with the team and GTA by the clients. The board includes a project backlog The team lead will take items from the Backlog and assign them to the rest of the team members. Each team member will work on their assigned feature as well as work in tandem with anyone whose assignment will depend on their part. Front-end and back-end team members will confirm connection points on features in addition to data types and methods used in these features. Before testing combined features, the members will reach out to at least one other member to conduct a code review and ensure that the design works together. The team will be

working iteratively and will follow the agile scrum workplace methodology. Sprints will be one week and will end with a client meeting.

Each feature of the project will have its own branch on the GitHub page. These branches will be as a reference point for the other team members to see how things are progressing aside from the methods and datatypes used.

There will be multiple designated branches for testing that will contain a full working version of what is currently completed. Also, there will be a master testing branch that will be used to combine all features. Before pushing code to the master testing branch, the feature must be confirmed to be in working order with no fatal or application breaking bugs. After a group member has pushed their code to the master test branch, they will pull from the branch to ensure that their code has integrated well completed code, with no system breaking bugs.

Bugs in the code that do not pose a threat to the entire system do not need to be handled before pushing to the master test branch. However, they must be well documented with the features, methods, and order in which methods are called so they can be addressed at a later date.

We will use Jenkins as our automation server to create a pipeline to GitHub that helps us continuously merge our as we go through the iterative incremental process.

1.7 Technologies

All of the technologies below have been requested by the client:

Backend:

- Google Firebase: Firebase provides a number of backend tools for mobile application developers such as Authentication and Realtime Database.

Frontend:

- XML: XML is a straightforward, easy to use language for frontend development with Kotlin. Android studio provides a drag and drop interface that generates XML code as the frontend is designed.

Testing and Integration:

- MockK: MockK was chosen because it is mocking framework built specifically for Kotlin.
- Jenkins: Jenkins is an automation server. We will be using it to implement continuous integration and continuous delivery of our code base. Jenkins was chosen because it is open source, widely used, and automatically triggers a build when a team member pushes their code into GitHub. This would allow the team to get support easily if needed.