CS 598 Senior Design 1
Work Statement

Team Name:
WuShock Go

Team Members:
Karishma Bhakta
Sriram Srinivasan
Fitri Rozi
Tan Tran
Dan Khuu

2nd Semester Work Statement
December 10, 2021

Introduction/Purpose/Objective

Tabor College is a private Mennonite liberal arts college in Hillsboro, Kansas, United States. Tabor has reached out to WSU regarding their ongoing problems and proposed a one-stop app for their campus. The purpose is to build a one-stop app for Tabor Students. Providing access to information on and about TABOR College, several campus offices, departments, divisions, and other resources and services that are offered. The objective of this application will ensure that students have a one-stop solution for them to access all features with the push of a button.

This work statement will lay out an outline of the Scope, Work Environment, Tasks/Deliverables, and Acceptance Criteria for our mobile application.

Scope

In order to complete this Statement of Work, team WuShockGo will be responsible for the planning, implementation, and testing of the mobile application. Our team will begin the development from the first-semester prototype and will work towards achieving the goals laid out in the Tasks/Deliverables. Our team consists of Computer Science majors therefore we will be focusing on the software and debugging.

Work Environment:

The work will be primarily done virtually by team collaborations on our laptops and school computers (if needed) as our project is software-based. We will be using Visual Studio Code, a text editor, to write the code and push it to GitHub. For the frontend, React Native framework will be used to build a native application for IOS & Android devices. The framework comes with multiple libraries to build the user interfaces, without having to create them our own from scratch. For the backend, Firebase will be used to handle backend work.

Tasks/Deliverables:

Note: This is made through Microsoft Project therefore the program doesn't count weekends.

Task	Week	Mobile App Development	Total Days: 184 Days	Fri 8/20/21	Wed 5/4/22	Description

31		Design UI / UX Design	20 days	Sat 11/13/21	Thu 12/9/21	Finalize prototype and send to sponsor for feedbacks
32	**∞**	**WINTER BREAK**	**∞**	**∞**	**∞**	**XMAS AND NEW YEAR**
33	**2 Weeks**	**Troubleshooting**	**14 days**	**Fri 12/10/21**	**Wed 12/29/21**	**Touch up on possible changes on prototype or any necessary adjustment**
34		Touch up on possible changes on UI/UX	3 days	Fri 12/10/21	Tue 12/14/21	Add any new or update
35		Team Discussion about Design 4 days		Wed 12/15/21	Mon 12/20/21	Finalize with team discuss thoroughly
36		Solve any Potential Issues	7 days	Tue 12/21/21	Wed 12/29/21	Solve any possible problems/issues that might occur with the design
37	**1 Weeks 4 days**	**Testing & Deployment Prototype**	**11 days**	**Mon 12/20/21**	**Mon 1/3/22**	**Testing phases where we test our prototype**
38		Send prototype to Sponsor	3 days	Mon 12/20/21	Wed 12/22/21	Send to Sponsor for feedback and agreement
39		Get Feedback and changes From sponsor	9 days	Wed 12/22/21	Mon 1/3/22	Ask for Sponsor Feedback and what needed to be added on the app
40	**17 Weeks**	**Development**	**120 days**	**Sat 10/23/21**	**Thu 4/7/22**	**Development Overview the whole entire Project**
41	**17**	**Front-End**	**120 days**	**Sat**	**Thu 4/7/22**	**Front-end start and**

	Weeks	**Development**		**10/23/21**		**end with Development**

42	**17 Weeks**	**Multi-Platform Development**	**120 days**	**Sat 10/23/21**	**Thu 4/7/22**	**Multi-Platform start and end with Development**
43		IOS & Android	90 days	Sat 10/23/21	Thu 2/24/22	React Native, expo CLI (command line interface)
44		Navigation (MENU + BUTTON)	90 days	Sat 10/23/21	Thu 2/24/22	Make sure that navigation and buttons function as it should be
45		DISPLAY	90 days	Sat 10/23/21	Thu 2/24/22	Balance, proximity, contrast, color
46	**5 Weeks**	**Testing**	**30 days**	**Thu 2/24/22**	**Wed 4/6/22**	**We will be testing the applications making sure it goes through various tests before deploying**
47		Internal Testing & Verifying	30 days	Thu 2/24/22	Wed 4/6/22	Do internal testing before sending it to the sponsor. Make sure all requirements are met
48		Feedback & pre-approvement	30 days	Thu 2/24/22	Wed 4/6/22	Get feedback & and changes. Make sure it gets approval from the sponsor.
49	**17**	**Back-End**	**120 days**	**Sat**	**Thu 4/7/22**	**Back-End start and**

	Weeks	**Development**		**10/23/21**		**end with Development**
50	**15 Weeks**	**Database**	**90 days**	**Sat 10/23/21**	**Thu 2/24/22**	**Database start with Back-End Development**
51		Set up Github Repository	7 days	Sat 10/23/21	Mon 11/1/21	Set up GitHub to store code and back up. Transfer to Tabor after project completion
		Web scraping script and storing information in Firestore	90 days	Fri 12/10/21	Thu 4/7/22	Web scraping research and implementation
		Firebase cloud functions to run scripts	25 days	Thu 1/20/22	Wed 2/23/22	Setting up cloud functions to run scripts that interact through Firebase API
52		Define Database Platform	7 days	Sat 10/23/21	Mon 11/1/21	Back-end members will define which database platform they are going to be using
53		Set up & Configuring firebase/firebase push notification	30 days	Mon 11/1/21	Fri 12/10/21	Begin set up and initial set up of project
		Firebase push notifications	90 days	Fri 12/10/21	Thu 4/7/22	
54		Testing	30 days	Fri 12/10/21	Thu 1/20/22	Back-end testing making sure of security flaws/database can handle multiple

						request
55		Debug	25 days	Thu 1/20/22	Wed 2/23/22	Check for any issues or errors when merging w/ back-end. Or when web scraping
56	**3 Weeks**	**Deployment/Documentation**	**20 days**	**Thu 4/7/22**	**Wed 5/4/22**	**Final Product sent out to Sponsor**
57		Deploy to Tabor & Andy	5 days	Thu 4/7/22	Wed 4/13/22	Get Professor Andy suggestions and approval before deploying to Tabor
58		Sponsor Feedback/improvement	15 days	Wed 4/13/22	Tue 5/3/22	Wait for approval and feedback then make changes
59		Write documentation on Github	5 days	Wed 4/13/22	Tues/5/3/22	Provide a READMe file and walkthroughs of how to set up the development environment

Work performance

Our team will meet once a week for SCRUM to satisfy Agile methodology. We will use Trello to collaborate with team members and manage tasks. The focus of the meeting is to summarize what we have accomplished and outline any issues that are blocking progress/development.

Acceptance Criteria

- Testing
- Hosting
- Deployment
- Google Play Store/ App Store

User Interface Testing (Front-End):

1. Documentation Testing
2. Functional Testing
3. Usability Testing
4. Performance/Compliance

Define:

- Documentation Testing
    - Documentation testing is part of the non-functional testing of a product. It is the necessary preparatory stage of the beginning of mobile testing.
        - We want to make sure to analyze the requirements given by the sponsor and screen layout, navigation layout, and other requirements that may obscure the design. Therefore we will analyze for any completeness and discrepancy.
            - Any discrepancies are required to be resolved before the development phase.
        - Test plan and Test case are needed to be defined or any other specifics.

- Functional Testing
    - Functional testing is aimed to ensure that it is working as expected and follows the requirements.
        - We checked whether the application performs the expected functions.
        - Installing and running the application

        - Installing the application should take place without significant errors if the device meets the requirements.
        - Verify that the application starts correctly
        - Ensure that there is a user manual available for the user.
        - Ensure that startup and exit procedures meet requirements.
    - Constant users feedback testing
        - Appropriate reaction of buttons while pressing.
        - Make sure notifications are working as directed.
    - Update Testing
        - Make sure user data is saved after an update.
        - Ensure update progress is displayed properly.
    - Other
        - Error messages display correctly. Either the page doesn't load or is down.

- Usability Testing:
    - Intuitive user interface that meets accepted standards. It's done to make apps quick and simple.
        - We want to make sure it performs well to create fast and easy-to-use applications
            - Satisfaction
            - Efficiency
            - Effectiveness
        - We would also like to consider the following in our usability testing to ensure that UI meets the requirement.
            - Button size and placement on screen
            - Module's navigation
            - Simplify the app's icon and images
            - Color and Text should be readable, clear, and visible.
            - Screen resolutions & responsiveness.

4. Performance/Compliance Testing:
    - Performance testing is a set of types of testing, to evaluate how the applications will perform in terms of responsiveness and stability under a particular workload. We want to examine the speed, robustness, reliability, and application size.
        - Load Testing
            - Testing the response time of the application to various requests.

        - Web scraping loading information / how it will populate on the app (time)
    - Stress Testing
        - Testing the application to see if it can handle multiple requests
    - Stability Testing
        - Examine the applications working during a long period of time under normal load
- Compliance Testing, making sure that the standard and policy are followed. Aiding in adherence to policy, rules, or regulation.
    - User access right / Terms and Conditions
    - Documentation Procedures
    - Software license (Sponsor's Optional)

User Testing (Back-End):

1. Security Testing
2. Continuous Integration / Continuous Deployment Testing
3. Unit Testing

Define:

1. Security Testing:

- Security testing is a type of testing to find any vulnerabilities and possible loopholes and weaknesses that may cause a loss. Its primary goals are to identify any threats in the system and help expose any possible security risks.
    - API Testing
        - Since we are using API we want to make sure to test its functionality, reliability, performance, and security
            - Functionality
                - Making sure it's working as intended. That it gives the correct output and connection
            - Reliability
                - Making sure how often the API endpoints experience any downtime
                - Latency, how slow for fast it responds to request
            - Performance
                - Determining stability and responsiveness of application when given a certain workload
            - Security

        - Making sure the conditions for user access, encryption and authentication concerns are met
    - Functional Testing
        - Confirms the application is working in line with predetermined requirements
    - UI Testing
        - Tests the efficiency and usability of the front and back end
    - Security Testing
        - Penetration Testing: Identifies weaknesses and loopholes in an application
            - where we try to find a vulnerability in the application
        - Fuzz testing
            - The goal of this test is to trigger crashes, infinite loops to detect any known and unknown vulnerabilities
    - Load Testing
        - Tests the application to see how it behaves when being accessed by multiple users synchronously
        - Ensures stability and smooth functioning

2. Continuous Integration / Continuous Deployment Testing:

- Continuous integration is a process where new code changes to an app are regularly built, tested, and merged into a shared repository. It is an optimal solution when there are too many branches of an app in the development stage that may create a conflict with one another.
- Continuous deployment is a process where a developers' changes in the code are pushed automatically to the user. It saves time in overloading operations teams with manual deployment processes that may slow down the app delivery.

    - Continuous Integration:
        - We can use GitHub to serve this purpose where everything is consolidated into one repository
        - Ensures that every code file belongs to each relevant section, enabling easier testing
        - The backend code written in JavaScript can be integrated and updated constantly as and when significant changes are made

    - Continuous Deployment:

        - Pushing changes and committing them often will result in the latest code always being used as the working version
        - Web scraping loading information / how it will populate on the app (time)
        - Enables us to test to a real-group of users and getting their feedback before deciding which areas to modify on

3. Unit Testing:
    - Unit testing is a process where individual mini components of the code are tested. It validates that each unit of software performs as expected. They consist of 4 steps: create test cases, review, set up a baseline, and execute test cases
        - Create test cases
            - Test cases are created with sample inputs that are similar to real-life data
        - Review
            - If the test data does not perform as expected, then review them and make any changes
        - Set up a baseline
            - Tests every single line of code in isolation without considering other functions of the code that incorporate that single line of code
        - Execute test cases
            - Have one or more test cases in the backend functionality so that app developers can ensure that any input successfully passes through the app
            - Ensures that the functionality and code modularity is maintained

Customer Feedback

Before the mobile app gets deployed to Google Play Store and App Store, participants from Tabor College will test the app and provide feedback so that developers can make changes to improve customer satisfaction. Once the app is ready and gets deployed, we can use the ratings and comments from the mobile stores to get a better assessment with larger sample sizes.

I have read the entire report and it meets my personal quality standards.

(Signatures)	(Names)
Karishma Bhakta	Karishma Bhakta
Sriram Srinivasan	Sriram Srinivasan
Fitri Rozi	Fitri Rozi
Tran Tan	Tan Tran
Dan Khuu	Dan Khuu