Scope and Charter

Offline First Real Time Bus Tracking
University Of British Columbia Okanagan
COSC 499

Authors:

Document Owner(s)	Role	Contact
Wasek Habib	Product Manager	wasek.edu@gmail.com
Matthew De Leeuw	Integration Lead	mattdeleeuw@hotmail.com
Trevor Gallicano	DevOps Lead	trevorg@hotmail.ca
Ini Oladosu	Technical Lead	inioladosu@gmail.com
Kyle Rennie	Client Liaison	kyrenzie@gmail.com

DOCUMENT HISTORY

Date	Document Version	Document Revision Description	Document Author
Oct 2nd 2018	1st Draft		

1. Identification

2. Problem Statement/Project Objective

- **2.1 Project Purpose**
- **2.2 Project Objectives**
- **2.3 Success Criteria**
- **2.4 Scope Statement**

3. High level Requirements:

- **3.1 Functional**
- **3.2 Non-functional**
- **3.3 Technical**
- **3.4 User**

4. Work Breakdown Structure

5. Assumptions and constraints

6. Environmental Constraints

7. High level risks

8. Methodology

9. Project Milestones

10.UML USe Case

11. Approvals

1. Identification

Name of Project: Real-time Bus Tracking App

Sponsor: Fremtid Media

Project Stakeholders:

Stakeholder Names	**Roles**
Fremtid Media	Sponsor/Client
Reza Afzali	Client
Simranpal Bains	Client
Scott Fazackerley	Instructor/Supervisor
UBCO Students/ Residents of Kelowna	Users
COSC Capstone Team 12	Android App Developers
Engineering Capstone Team	Hardware/Infrastructure Team
University of British Columbia	Academic Institution
BC Transit	Open Data Provider
FirstCanada	Kelowna Public Bus Transportation Operator
Kontakt.io	Beacon Provider
Google	Android OS Provider
Here	Maps API Provider

Team:

Team Members	**Roles**	**Responsibilities**
Wasek Habib	Product Manager	- Primarily responsible for the project workflow, requirements engineering, software architecture, and ensuring SDLC. - Involved in design and development process.
Ini Oladosu	Technical Lead	- Responsible for the non coding documents - Responsible for UI Design of application - Involved in development process
Matthew De Leeuw	Integration Lead	- Responsible for checking code before integration into master branch (aka) responsible for what gets deployed - Responsible for programming with team on seperate features
Trevor Gallicano	DevOps Lead	- Responsible for continuous integration/deployment - Responsible for automated testing - Involved in development process
Kyle Rennie	Client Liaison	- Responsible for continuous client contact and logging conversation - Responsible for communicating clients wants

2. Problem Statement/Project Objective

2.1. Project Purpose:

Currently there is a ridership of 4.9 million in Kelowna per year, a fair amount of riders (>1%) use the transit app, with this app we are looking to take a portion of the demand for maps and transit for this app. Current apps have issues with telling the user the real time of their bus arrival because they are unreliably crowdsourced, or just use the bus schedule for times. This app will make transit passengers be better notified about their bus whereabouts and when they should be leaving to meet the busses arrival at an optimal time. They will be able to view the exact whereabouts of the bus on the map and get notified when a bus is on its way with its expected arrival time and they will also be notified when a bus as left the bus stop if they are not on it. If the project is not completed on time then the company overseeing it can chose to continue with a new or the same team. They would have prototypes and documentation to help them continue with the project, even with a new team.

2.2. Project Objectives:

Create a mesh network with provided beacons and then create an app that draws information from the network. The app should notify users with a busses expected arrival time and when the bus has left the stop if they are not on it. They should also be able to track a busses exact location when in the mesh network.

2.3 Success Criteria:

- The app will have simple and elegant user experience following standard human computer interaction rules
- User satisfaction will be measured by a standardized set of questions followed by the usability test and will measure 3 or higher on a 5 point scale.
- The user interface will follow the best UI practices and ‘material design’ standards.
- Response time should not be more than 2-3 seconds.

2.4 Scope Statement:

The offline first android app (proof of concept) will show the user live location and arrival time of the bus (real-time bus schedule) in nearby bus stops. The user will get a push notification of the remaining arrival time of the subscribed bus number when he/she is in a certain range of the beacon at the bus stops. The beacons will be installed and configured by the engineering team, which create a mesh network. The application will receive the bus/car information through Bluetooth Low Energy (BLE) provided by the mesh network. If the user clicks on the push notification, it will take him/her to the app and show the live location of the bus on the map. The users will not need an account to use the application. Personal information of the users will not be asked or stored. However, their travel time and distance will be stored anonymously.

Out of Scope:

- Outside of UBCO roundabout and EME bus stop.
- Setting up the beacons, checking physical conditions and health.
- Creating mesh network.
- People with strollers or wheelchairs (physically impaired).
- Setting up and working with speakers and displays on bus-stops.
- Any communication between speakers, beacons, and light sensors.
- Any kind of hardware implementation. The engineering team is responsible for that.
- Cost analysis
- Wayfinding (Navigation and in-between stops).
- Anything that is in scope of the engineering team.

3. REQUIREMENTS

3.1. Functional Requirements

- System can connect to the beacon mesh network
- System displays a map with the bus locations
- System will ask for user's permission to access to GPS, bluetooth, and internet
- System tracks the live position of busses
- Bus tracking can be done while the device is offline
- System displays the expected arrival times of busses
- User receives a push notification when they are within a set distance of a bus stop beacon.

- User is notified about the wait time for the next arriving bus
- Application will be available to all users with no registration
- Application shows a list of nearby bus/car(s) that has/have beacon installed
- App tracks the current location of a user in the background
- App tracks travel distances and travel times of a user
- Travel data is encrypted while stored on the device
- Travel data is sent to an online database when the device has an online connection
- Travel times will be deleted from device after they are sent to an online database
- Admins can view data stored in the online database

3.2. Non-Functional Requirements

Development

- Developed as an Android application
- Requirements and design will be developed using the waterfall method
- Development and testing will be completed using the agile method
- Application will use third-party API's
- Costs for development (such as cloud service or third party API's) will be covered by the client

Performance

- User must be outdoors on the UBCO campus and within range of the mesh network
- Software will handle number of users equal to all bus commuters at UBCO
- Software will be scalable up to the entire 97 bus route in Kelowna
- Bus tracking will represent bus location accurate within 100 meters
- Bus position will be updated within 3 seconds (including screen latency)
- System requires all beacons to be functioning
- System downtime limited to time spent replacing beacons
- Bus tracking will be done with no internet connection
- Bus tracking will only show information related to the next bus(es)
- System can track any bus (with a beacon) within the mesh network
- Travel times will be stored on device while the device is offline

3.3. Technical Requirements

Tools/Languages/Libraries	**Type**	**Version**
Android Studio	Development Environment	
Balsamiq	Wireframing	
Adobe XD	Mockup	
Docker		
Jenkins/Travis	CI Solution	
Git	Version Control System	
Github	Version Control Hosting Service	
Trello	Project Management	
Slack	Communication	
SQLite	Relational Database	
Couchbase Mobile		
AWS/Firebase		
Java		
Node.js		
Google/Here/Mapbox Maps	Maps API	

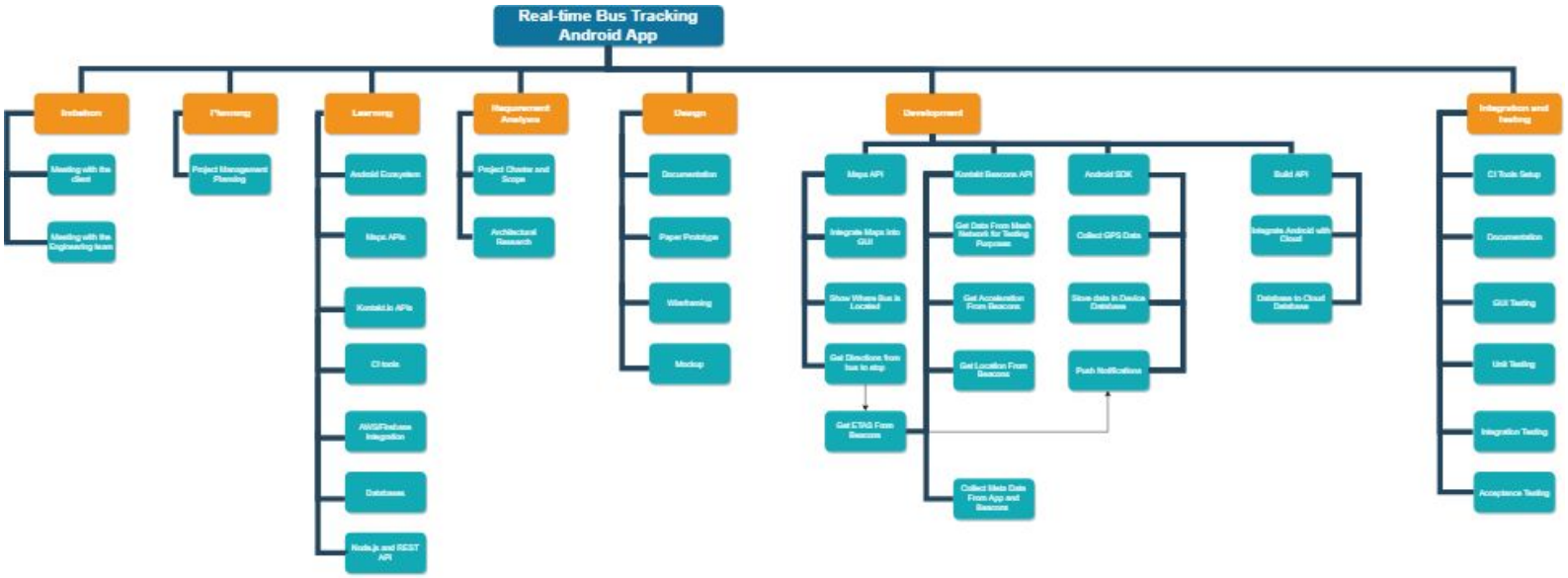
- System can connect to the beacon mesh network
    - Use Kontakt Beacon APIs
    - API key will be stored in the app

- System displays a map with the bus locations
    - Use Here maps API and cache the latest location
    - API key will be stored in the app
- System will ask for user's permission to access to GPS, bluetooth, and internet
    - Use AndroidManifest.xml file to ask for user permission
- System tracks the live position of busses
    - Use Kontakt Beacon APIs and Here maps APIs
    - API keys will be stored in the app
- Bus tracking can be done while the device is offline
    - Use Kontakt Beacon APIs and Here maps APIs
    - API keys will be stored in the app
- System displays the expected arrival times of busses
    - Use Kontakt Beacon APIs and Here maps APIs
    - API keys will be stored in the app
    - Store beacon UUIDs and geo-coordinates in the database
- User receives a push notification when they are within a set distance of a bus stop beacon.
    - Use async push notification
    - Track user location and desired bus location in the background using GPS and Kontakt Beacon APIs
- Application shows a list of nearby bus/car(s) that has/have beacon installed
    - Use Kontakt Beacon APIs and BLE
    - API keys will be stored in the app
- App tracks the current location of a user in the background
    - Track user location and desired bus location in the background using GPS
- App tracks travel distances and travel times of a user
    - Use Here maps APIs to calculate the distance and time
    - API keys will be stored in the app
- Travel data is stored on the device anonymously
    - Use instance id
- Travel data is sent to an online database when the device has an online connection
    - Use /POST API

3.4. User Requirements

- Use application without an account
- User receives push notification when entering specified beacon range
- Users can track bus with map
- User can see expected bus arrival time

4. Work Breakdown Structure



Task List	Owner	Estimated Hours					Total Estimate
Learning	All	Ini	Kyle	Matt	Trevor	Wasek	Average Estimate
Android			15	30	20		
Maps			2	3	3		
Kontakt.io			3	3	3		

CI Tools	Trevor		2	16	10		
AWS/Firebase			2	4	3		
Databases			2	4	3		
Node.js			5	4	5		
Design	All						
Documentation			15	4	6		
Paper Prototype			4	0.5	1		
WireFraming			2	1	1		
MockUp			2	2	2		
Development							
Maps API			2				
Integrate Maps Into GUI			3		3		
Show Where Bus is Located			3		3		
Get Directions from bus to stop			1.5		2		
Kontakt Beacon API			2				
Get Data From Mesh Network for Testing Purposes			4		3		
Get acceleration from beacons			3		1		
Get locations from beacons			4		1		

Collect meta data from app and beacons			4		3		
Android SDK							
Collect GPS Data			3		2		
Store GPS Data					3		
Build API			5				
Integrate Android with Cloud			4		4		
Local Database to Cloud Database Path			3		3		
Integration and Testing							
Ci Tools Setup			1		2		
GUI Testing			2		8		
Documentation			3		6		
Unit Testing			2		10		
Integration Testing			4		8		
Acceptance Testing			5		5		

5. Assumptions

- UBCO will approve installing beacons on campus.
- A beacon will be set up in one bus upon approval from BC transit. If not, the students will use their own transports. The engineering team is mainly responsible for this.
- The engineering team will be responsible for installing and configuring beacons and will provide any help needed when COSC team test them.
- The beacons given will work and respond as expected.
- The developers have knowledge of Java and relational databases.
- Third party APIs (i.e. maps API, beacon API) will work properly..
- The team built APIs will be public and will not ask for any authentications.
- The cloud service will have at least 99% uptime.
- The beacon range will be 50-70 meter.
- The beacons will not be stolen or vandalized.
- The team will be agile, self-organized, and maintain synergy.
- Any expenditure including cloud service, third party APIs, gas will be provided by the client.
- The communication between stakeholders will be prompt, clear, and efficient.
- Each developer will be working 6-8 hours average during school weeks.
- The developers will not work in December due to final exam and holiday.
- There will not be any requirement changes from the client after October 16, 2018.

6. Environmental Constraints

- Requires approval from UBCO and BC transit to install beacons on campus and bus.
- The durability and performance of the beacons in Kelowna weather is unknown.
- Limited in-person meeting with the client.
- Field testing will be challenging during hostile weather.
- No control over third party APIs and cloud service.
- Some tasks are dependent on the task completion of engineering team and can not be done parallel.
- Majority of the developers are not familiar with Android ecosystem, building REST APIs with node.js, and NoSQL databases.
- Lack of domain knowledge will lead to unrealistic estimations.
- Time constraint.

7. High Level Risks

Risk	Likelihood	Severity	Mitigation Plan
Unfinished Project	Medium	High	Plan to stay ahead of deliverables.
Conflict with Engineering Team	Low	High	Frequent communication and transparency.
Losing members	Low	Medium	Involve all team members in development so we can pick up where other left off.
Unmotivated Team Members	Low	High	Constantly staying ahead of deliverable due dates
Faulty Beacons	Medium	High	Have as many as possible so we are not reliant on just a few.
Client company shut down	Medium	High	Continue project with Scott.

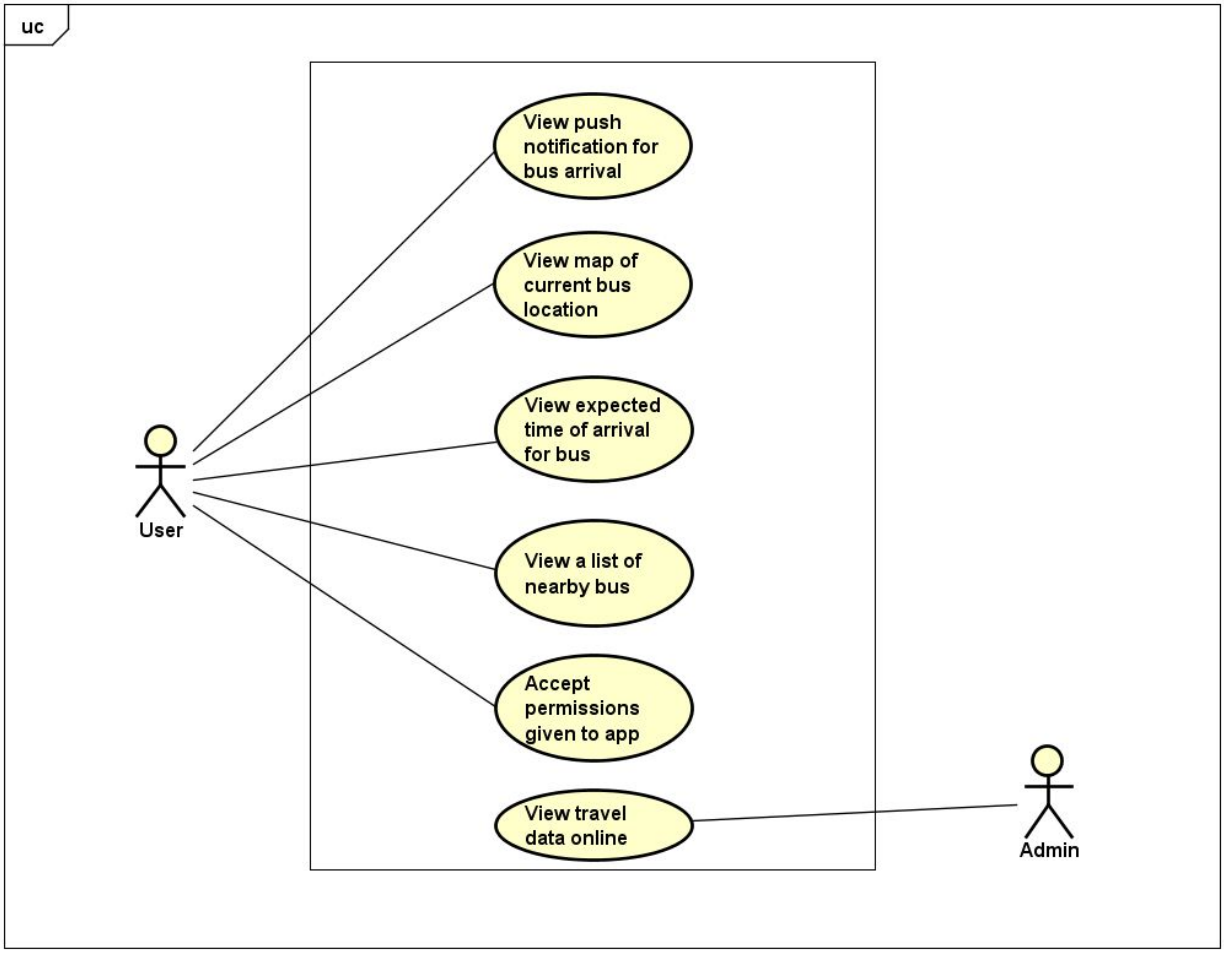
8. Methodology

Scrumban methodology and Trello will be used for project management. Weekly sync up, demo, planning meeting will be held on Monday. If Monday doesn't work out for some reason, the team will have meeting on Friday or Tuesday before the class (not recommended). The sprints will be one week long starting from every Tuesday. Backlog will be created in each week's planning based on the progress and priority. Every week the developers will pick up the tasks from backlog they think they can finish in that week. If a task seems big, sub tasks can be created under that task. Everyone is expected to finish at least 80% of their workload each week, although 100% is preferred. The rest can be carried forward to the next week's sprint. There is no need for daily scrum meeting as trello is synced to the scrum channel. So everything knows who's working on what.

9. Project Milestones / Deliverables

Milestones / Deliverables	Due Date	Complete (YES / NO)
1st Meeting with Scott	October 2nd, 2018	YES
Scope and Charter Document & Presentation	October 16th, 2018	NO
Design Requirements Document & Presentation	November 13th, 2019	NO
End of Semester Deliverable	TBA	NO
MVP Presentations	January 8th, 2019	NO
Final Deliverable	TBA	NO
Final Project Presentation	April 8-26	NO

10. UML Use Case



11. Approval

Project Manager:_____ Signature: _____

Project Sponsor:_____ Signature: _____

Report Expectations (80%)

- (5%) Project charter
- Scope Statement Document including
    - (5%) A description of the software you are building
    - (5%) A precise set of the criteria that your client deems as successful outcomes of your project
    - (10%) A list of functional requirements
    - (10%) A list of non-functional requirements
    - (5%) A list of technical requirements
    - (5%) A list of user requirements
    - (2.5%) A list of environmental constraints
- (2.5%) A description of your proposed workflow and methodology (what is the development process/how will it work)
- (10%) Identification of user groups and a complete set of UML use case diagrams (based on your initial assessment)

(20%) A simplified work/feature breakdown structure (see below for details)