Discovr - Requirements, Vision, and Scope

By Simon Jinaphant, Yueyue Zhang, Valerian Ratu, David Wong, Nicholas Leung, Jacqueline Choi

Product Vision Statement

Discovr offers a convenient and immersive experience for those who spend their days on UBC's Vancouver campus. At its core it provides a graphically interactive map containing relevant information that cannot be easily found or accessed from a search, such as obstruction aware pathfinding and local event notifications. Discovr's convenience will help users plan out their day by making them more aware of what is around them on campus.

Problem Statement

The problem of	1. Scattered resources regarding campus events and information 2. Unawareness of the blocked zone or constructions within UBC campus
affects	The UBC Vancouver community
The impact of which is	1. A lack of attendance and participation of community events around campus 2. Being forced to take detours making people late for classes, labs, and meetings
A successful solution would be	A graphically interactive map with relevant information that cannot be easily found or accessed from a search. It helps students plan out their day by making them more aware of what is being offered around campus. As well, it is a convenient and organized location to discover events and campus information

Product Position Statement

For	All UBC community members
Who	Want to get involved
Our System	Is all software
That	Can detect the construction at UBC and gather local event information
Unlike	The iPhone and android apps with slow and cumbersome UX
Our Product	Is always up to date and prioritizes events most relevant to the user

Summary

Users:
The main users of our program would be people who are at campus day to day: students, staff, and event organizers. They will use this application to check construction and blocked off areas on campus, or information on events happening on campus on that day.

Most students at are proficient at using smartphones and will expect our app to behave similar to most other apps on the play store. The user would also depend on the app to be robust and reliable and will not need additional information once the initial downloads (map and events) are set up - though with no connectivity they might not be as up to date.

Feature List:
- Personalize content by finding relevant information for YOU
- Directions with construction awareness
- Translink Bus schedules

Constraints:
- We have to use a lot of APIs, which increases the complexity of our app
- Our application has large dependencies from external sources
- Local data storage has to interact properly with remove database
- We will only consider developing for Android

Scope and Limitations
The scope of this program would only be the UBC Vancouver campus. Events/information beyond that will not be considered in the program. As well, we are ignoring iOS and a complete web-app users which limits the audience for this product within the campus.

Assumptions and Dependencies:
Expectations from other similar apps:
- The rank of regular events happening within campus.
- Notifications of the completion of selected constructions.
- If some area is blocked for a specific event, then it would be good to match them up

Dependencies from open sources:
- Map related data
- GPS system for locating user's current position

Use Cases

Use Case 1:
Goal: User wishes to see the list of events happening on campus
Primary Actor: User
Scope: Mobile App
Level: User
Precondition: User has app opened and loaded with internet connection
Success end: App loads a list of all events on campus
Failure end condition: App fails to load any event
Trigger: User selects the 'All Events' button on the app navigation

Main Success Scenario:

1. User opens the navigation drawer
2. System displays a navigation bar
3. User selects the 'All Events' option
4. System displays a list of events that are or will be happening on campus
5. User may select the event for more information - see Use Case 2.

Variation:
A. User wishes to see current and upcoming events on campus on the map

A3. User selects the 'Upcoming Events' option
A4. The system returns to the map and displays a list of pins on the map where events are located.
A4. The user may select a pin to find more information about the event - see Use Case 2.

Use Case 2:
Goal: User wishes to find out more information about an event on campus
Primary Actor: User
Scope: Mobile App
Level: User
Precondition: User is either on the All Events page or the Subscribed Events page or on the Map page with Event pins enabled
Success end: App moves the map to the location of the event and provides more information about the event
Failure end condition: App does not display more information to the user
Trigger: User selects a specific event

Main Success Scenario:

1. User selects an event
2. System obtains more information about the event
3. System brings the user back to the map where the event is located alongside an informational panel on with event information

Use Case 3:
Goal: User subscribes to an event

Primary Actor: User
Scope: Mobile App
Level: User
Precondition: User has taken all actions for use case 2
Success end: The Subscribed Events page now contains the selected event
Failure end condition: The subscribed events page does not contain the selected event
Trigger: User selects the subscribe button on the event information page

Main Success Scenario:

1. User selects the subscribe button on the event panel
2. System changes the subscribed button to inform the user they have subscribed to event
3. System moves the event to the Subscribed Events page

Variation - User unsubscribes from event:

4. User navigates to their list of Subscribed Events
5. User selects a specific event
6. System moves the event to the Subscribed Events page
7. User selects the unsubscribe button on the event panel
8. System changes the button to notify the use they are unsubscribed to the event
9. System removes the event from the subscribed events page

Use Case 4:
Goal: User wishes to upload their course calendar to the app
Primary Actor: User
Scope: Mobile App
Level: User
Precondition: User has their ical file in their Downloads folder
Success End: User has a list of their courses within the List All Courses page
Failure end condition: User does not have anything on their List All Courses page
Trigger: User opens up the List All Courses page on the app

Main Success Scenario:

1. User opens up the List All Courses from the navigation drawer
2. System finds the ical file within the user's Downloads folder
3. System analyzes this ical file and stores courses within the phone for future use
4. System displays the List All Courses page with the user's course information in it

Use Case 5:
Goal: User wishes to find out more information about their course
Primary Actor: User
Scope: Mobile App
Level: User
Precondition: Use has successfully imported their courses into the app
Success End: User obtains more information about their courses

Failure end condition: User is not able to get information about their courses
Trigger: User selects a course from their List All Courses page

Main Success Scenario:

1. User selects a course from the List All Courses page
2. System searches within the database for more information about the course
3. System moves the map to the building of the course
4. System provides an information panel about the details of the course

Extensions:

3a. Course has no associated building

3a.1 System maintains the map at it's current location
Proceed to 4.

Use case 6:

Goal: Planner wishes to submit their event into the app
Primary Actor: Planner
Scope: Website
Level: Planner
Precondition: Planner successfully accessed the Discovr event webpage
Success End: Planner successfully submitted their event into the app's database
Failure end condition: Planner couldn't submit their event into the app's database
Trigger: Planner navigates to the Discovr event webpage

Main Success Scenario:

1. Planner accesses the Discovr event webpage
2. Planner inputs information on the fields of the webpage (event name, location, time, description)
3. Planner selects Submit Event on the page
4. System inserts the event into the database and returns to the planner a unique GUID associated with the event

Extension:

A. Planner wishes to update their event information

A.1 Planner accesses the Discovr event webpage
A.2 Planner enters the GUID associated with the event
A.3 System finds the event and returns them back a page with their event information
A.4 Planner updates their event information and submits
A.5 System pushes the update into the database

Non-functional Requirements

Performance:

1. The app should not drain a huge portion of the user's battery and battery usage should be comparable to other similar apps like Google Maps

2. The app should try to reduce network usage when loading information from remote servers, especially when using mobile data

Safety:

1. Application should notify users to be aware of surrounding conditions and give details to the UBC safewalk program when using map features past sunset hours

Security:

1. Secure the API Key for for both Mapbox and Translink to prevent external users from using up all of our API transactions
2. Secure remote database connection string to prevent malicious access to our data.

Software Quality:

1. The app must be reliable and robust such that the user can depend on it on their daily workflow
2. The app must be simple and straightforward such that information can easily be found and doesn't have a high requirement from the user

Changes From Original Requirements

- **Event Tagging**
    - Event tagging was removed from the application as the complexity was too great for its purpose. Users can customize their experience by subscribing to events or searching for events instead.
- **User Profile**
    - As event tagging was removed, the need for a user profile to store tags was unneeded. Rather, the application only needs to store the .ical file of a user which can be done without a designated profile.
- **Nearby Attractions / Services**
    - This feature was removed mainly from the lack of time and due to redundancy. When buildings are shown on the map, the names of nearby buildings also appear and can be selected.
- **Construction aware pathfinding**
    - Due to Mapbox not offering any way to customize pathfinding in order to avoid certain areas, this feature cannot be included within the remaining time we have left.