Table of Contents

1. Introduction

1.1 Project Summary

Laze is an application containing a live and dynamic GPS map overview of Wilfrid Laurier University. It allows users to pin points of interest, such as study spaces, restaurant lines, parking spots, etc - all of which is visible to all other users of the application. These pins enables users to view indicated information such as the length of a line or the availability of a parking spot nearby. Users have the ability to support the validity of an active pin or flag to decrease its time before expiry.

1.2 Purpose

This document will cover all of the requirements needed for Laze and will describe in depth its features and how they work. The intended audience for this document is the client along with the rest of the project team. This document should be used by the project team as a guideline.

1.3 Scope and Business Model

1.3.1 Scope

This document demonstrates the properties and constraints of Laze, an application which shows users publicly posted live points of interest around Wilfrid Laurier University campus. Laze aims to provide free useful up-to-date live information to all users.

1.3.2 Business Model

Laze is a free-to-use application. There are no cost for users to use the web version of Laze. Our value proposition is to provide useful information to as many users as possible so we don't want to deter new users behind a paywall. Laze will also be void of any irrelevant advertisements or pop-ups. Instead revenue generation will be from sponsored promotions in the form of pins and live feed updates of small business or whoever wishes to purchase a highlighted/top spot for a period of time.

1.3.3 Business Example

Users have the ability to highlight a certain post they make on any of the corresponding tabs they choose to post in. For example, in the food tab, businesses such as Mozy's Shawarma have the ability to pay a one-time fee to list a promotional lunch offer that they are running. This allows for exclusive promotional offers to be well recognized, helping businesses maximize their visibility to users.

1.4 Definitions, Acronyms and Abbreviations

1.4.1 Acronyms and Abbreviations

- API - Application Program Interface - External software
- GPS - Global Positioning System
- GUI - Graphical User Interface
- POI - Point of Interest
- SRS - Software Requirement Specification
- TOS - Terms of Service

1.4.2 Definitions

- Pin - A visible marker placed on a map to indicate a point of interest
- Feed - Live scrolling chart of text, translating pins into text based on description

1.5 References

- This SRS document is based on IEEE Std 830-1998.

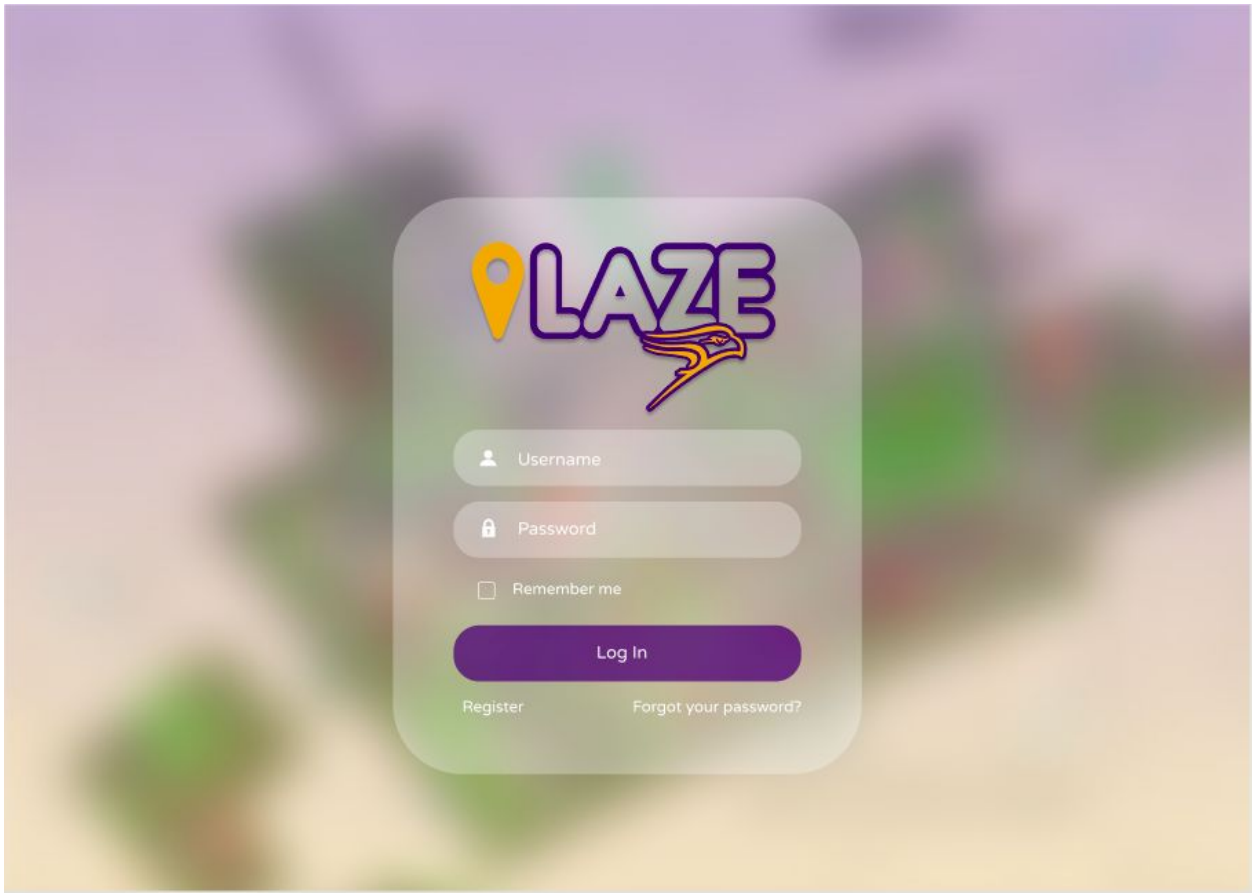
2. Overall description

2.1 Product perspective

Laze is an application based on a web platform, comprised of two major components, the front end and the back end. Students who use this app can input points of interest (such as open study spaces, restaurant lines, parking spots etc.) which can be viewed by other students. There is no need to install any prerequisite software to use this application.

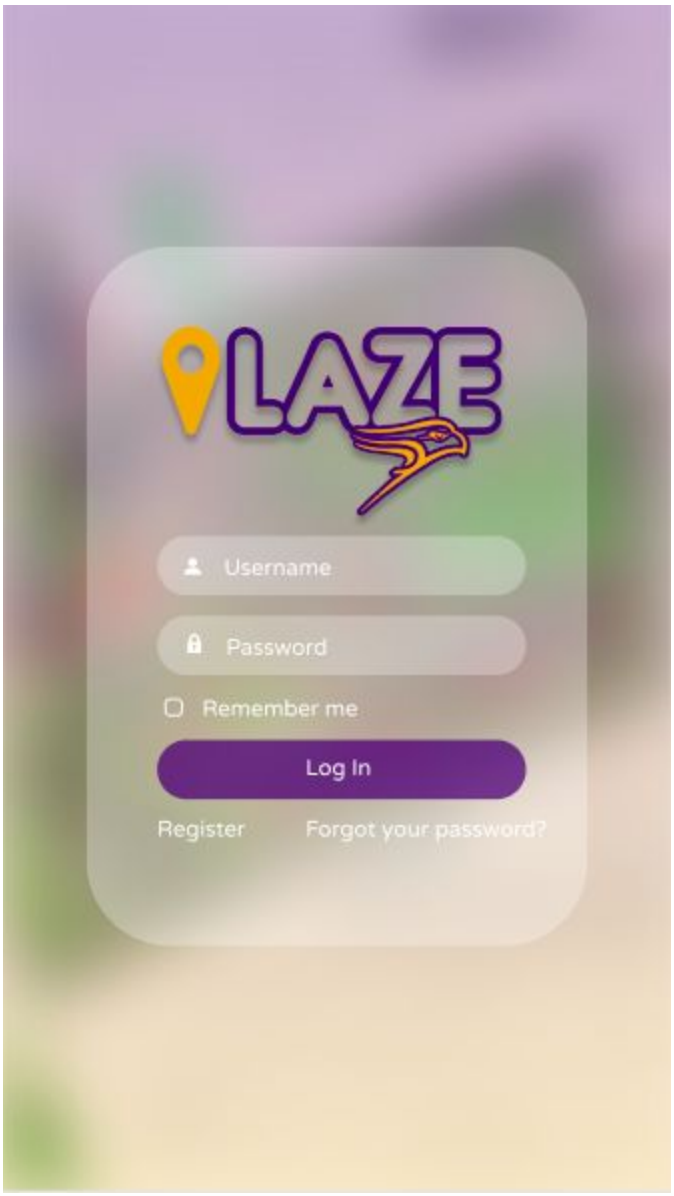
2.1.1 Sample GUI

Landing Page Browser



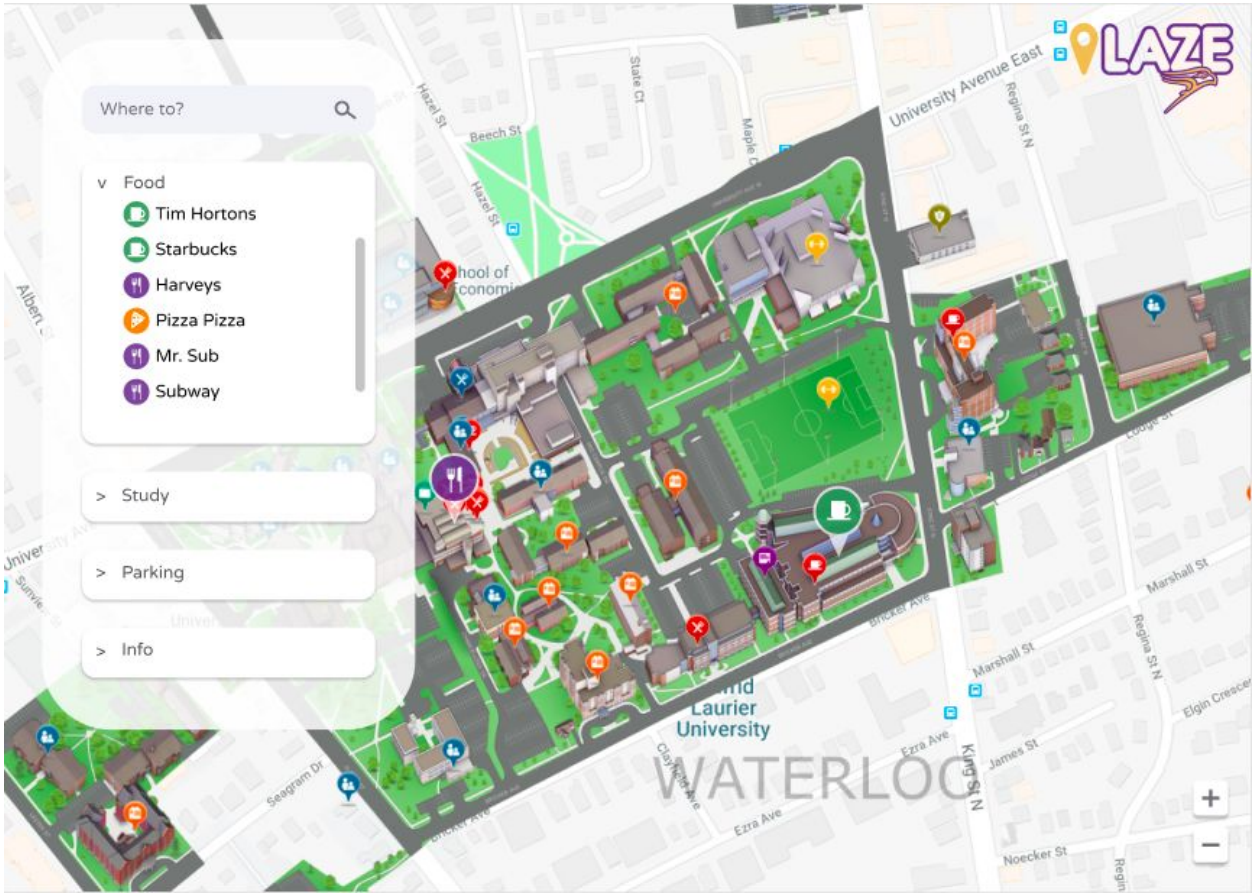
Above, there is a sample login user-interface for accessing Laze through the web. Guests have the ability to register and create a new account. Existing users can enter in their credentials to log in.

Landing Page Android / iOS (potential)



Above, there is a sample login user-interface for accessing Laze through a mobile device (Android or iOS). The graphics and colours used are consistent across all platforms.

Main Page Browser



On the main page, users can see an interactive map of Wilfrid Laurier University and the surrounding areas. Here, users can view pins and various points of interest such as restaurants and cafes. On the left, users can navigate the map by using the highlighted search feature to filter search results (such as "coffee" or "sandwiches")

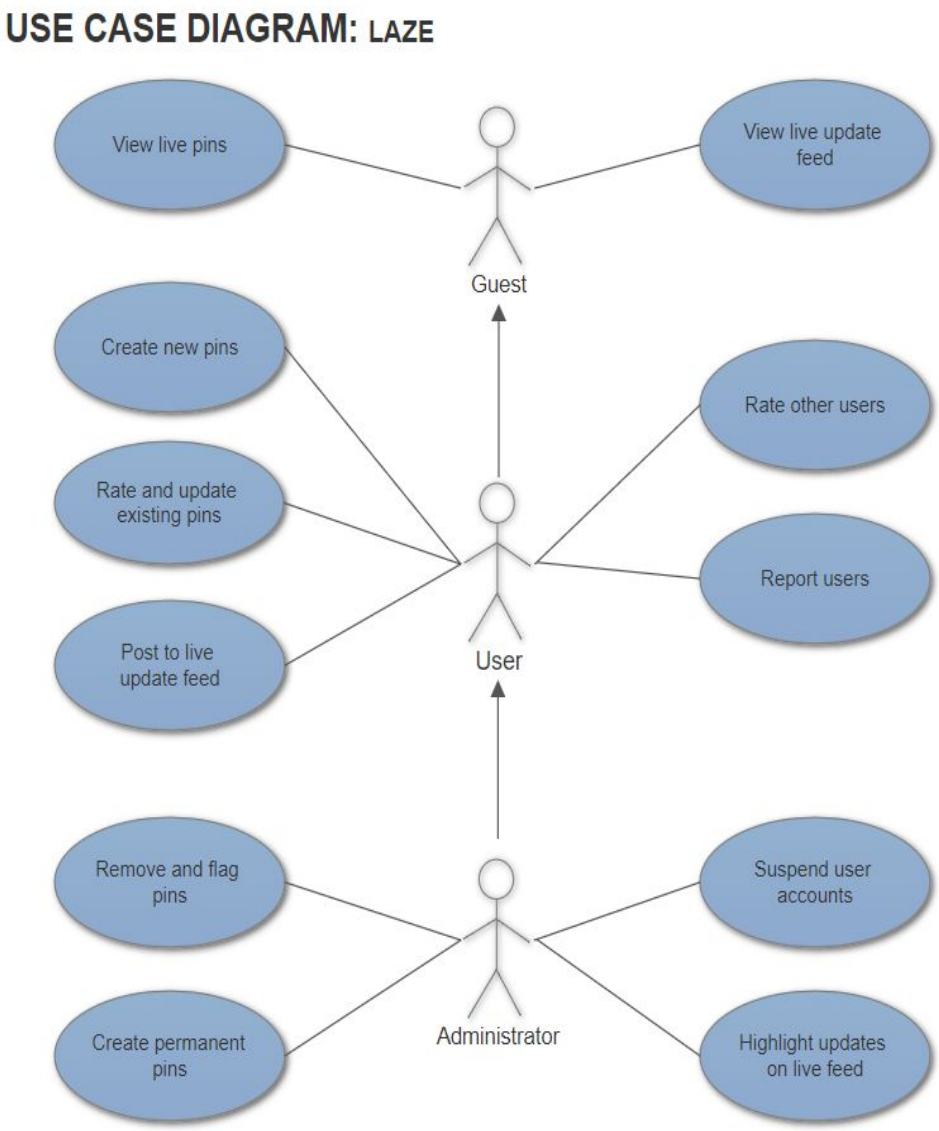
In addition, the left section also indicates what types of pins can be viewed for specific categories. For example, clicking the study tab opens a drop down list of all available pins corresponding to studying. These can include open study spaces or empty classrooms.

Users can zoom in and out of the static map using the + and - icons on the lower right hand side.

These sample user-interfaces serve as a basis for what our development team should strive for. All user-interfaces are subject to change as the project progresses.

2.2 Product functions / functional requirements

The users of Laze have the ability to notify other user of specific information about study spaces as well as places to get food etc. They are notified of this information through the map. If the user loses connection and is then reconnected, they will receive updates on all pins that they may have missed and that are still relevant to them. Additionally, Laze has the ability to filter out irrelevant pins that are not useful to the user.



2.3 User characteristics

The intended audience of Laze are those looking for more information about Wilfrid Laurier University Campus and surrounding areas. Users are expected to have a basic understanding of desktop and mobile application usage, but if they require further assistance, our information icons throughout our pages can guide them through it. Previous use of apps such as Google Maps, Uber, and Waze will assists in the amount of time required to become familiar with our application. The user interface is intended to be simple and more importantly, intuitive. Registration, pin placement and commenting will be no different than other application.

2.4 Constraints

The website has to be able to allow multiple users to access it at once. However if the user is not able to connect to the internet they will not be able to use the app as it relies on back end data to function. The user's must also be able to have access to google maps through the internet, because if google maps is down then the app will no be able to function as intended. There is also the development constraint of time, as this project is on a schedule and as everyone working on this project is student, there may be severe time constraints when attempting to meet project goals on schedule.

2.5 Assumptions and dependencies

It is assumed that users of Laze will have their location permission enabled and acceptance of our terms of service during user profile creation. Laze's TOS will indicate to users that misuse of the service is strictly prohibited and will result in banning of all accounts. Violations will include pinning of false information as well as malicious use of our data for 3rd party gain. Any attempts to alter, hack or collude with our software is also strictly prohibited.

Laze is also assumed to be functioning on a device with internet web browsing capabilities or application access capabilities. We assume users are familiar with their device's own capabilities and functions.

3. Technical Requirements

Italicized below are questions we aimed to answer throughout our process.

3.1 External Interfaces

- *What are our other dependencies? I.e. are we relying on Google GPS coordinates?*
    - No, our only dependencies are Google Maps API, we are not relying on any GPS coordinates.
- *What is our source for the live data? I.e. if a user wants to see the line at Tims, how will this be captured?*
    - The source for our live data is through a crowdsourced database.
    - *If it is being captured via live photos being uploaded by other users, do we need to consider privacy concerns?*
        - No, we will not be sharing any personal location or profile information.
- *What is our error logging mechanism?* - errors will be displayed in the server console log.

3.2 Performance Requirements

- *How many concurrent users do we have?* - In theory, there is no limit on concurrent users
- *What is our baseline? I.e. if we have 5k users @ a time, how will our application response time be?*
- *What is the availability of our app? Is it 24/7 or is there a downtime for maintenance?*
    - It is available 24/7
- *Resilience and robustness? I.e. if we load our app on one server only and it crashes, what is our disaster recovery?*

As of December 2018, we haven't hosted our app so some of these questions are not applicable.

3.3 Database Logical Requirements

- *What kind of database are we building? I.e. sequel, no-sequel, Hadoop, etc*
    - SQL
- *Is our DB read only or write as well?*
    - Read and write

3.4 Security

- *How secure is our application? I.e are we using HTTPS protocols?*
    - Not hosted
- *How are we writing logs?*
    - Only errors are being logged

3.5 Portability

- *What is the compatibility of our application? I.e. is the browser/mobile compatible?*
    - It works in browser
- *What are our accessibility standards? I.e. can a handicapped person use our application?*
    - No special features for disabled as of yet

4. Interface Requirements

4.1 Description

Due to the differing design restrictions and guidelines, the iOS and android versions may not be exact mirror interfaces. We aim to keep our styling consistent across all platforms and users should see minimal differences.

Version 1.0 - 30/09/2018

Created the document

- Ashaab Khan, Brayan Boukhman, Jay(YongJing) Li, Luke Krete, Mansi Bhatia

Version 1.1 - 09/10/2018

Adjusted document based on suggestions by David Brown and fixed formatting issues

- Binoli Joshi, Charmi Desai

Version 1.2 - 14/12/2018

Adjusted words and sentences used. Editing the document

- Zizheng Huang

Version 1.3 - 15/12/2018

Content reviewed and final copy pushed onto Github

- Mansi, Ashaab, Judy