

# Laze

## Table of Contents

---

- 1. Introduction
  - 1. Purpose
  - 2. Scope
  - 3. References
  - 4. Overview
- 2. Interface Description
  - 1. Module Interface
- 3. Detailed Design
  - 1. Module Detailed Design
  - 2. Coding Standards
  - 3. Documentation
- 4. Versions

## Introduction

---

### Purpose

This software design document states how the requirements will be achieved by describing the architecture and system design of the Laze web application. This document will provide descriptions of how each module will work including their associated methods and attributes.

### Product Scope

The scope of this project is to continuously provide live updating information for users around Wilfrid Laurier campus. Users will have the ability to set up their own data points as well as gain information from other user data points around them.

### References

IEEE. IEEE Std 1016-1998 IEEE Recommended Practice for Software Design Descriptions. IEEE Computer Society, 1998

CS312 - Software Engineering. [Software Design Document for Electronic Scrolling Displays](#). TGR Karthik, 2006

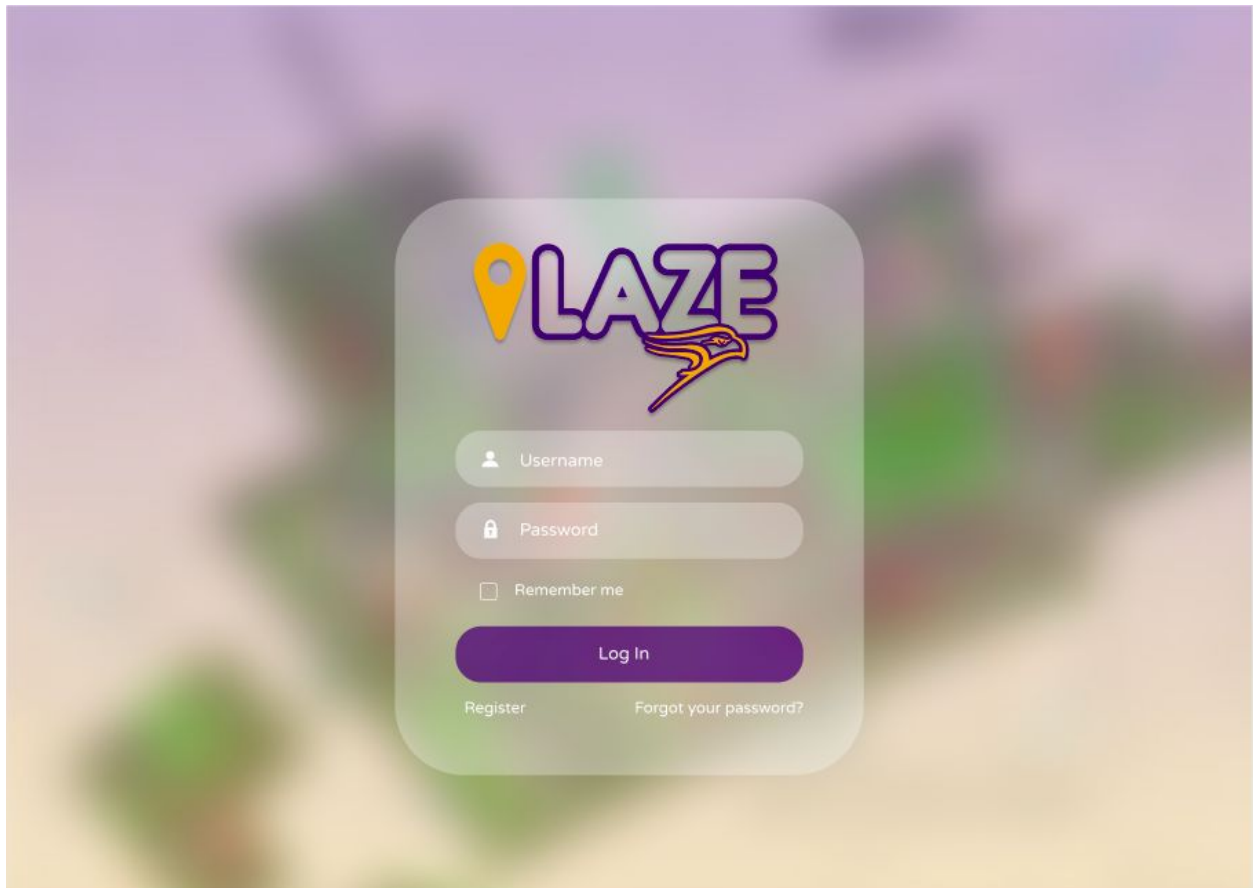
## **Overview**

This document is written following the guidelines of the IEEE Recommended Practice for Software Design Descriptions. It outlines the module, process, and data dependencies, as well as their detailed design and decomposition.

## 2. Interface Description

### 2.1 Module Interface

#### 2.1.1 LoginView



Field	Type	Description
username	input text	The user's login name
password	input text	The user's login password
rememberMe	button	Radio button user can select to save username.
login	button	Submits user credentials and logs them into MainView if they are correct.

register	button	Opens dialogue box that requests signup info
forgot-password	button	Creates dialogue with a textbox to prompt user to enter recovery email address.
guest-login	button	Allows the user to login as guest to view the MainView

### 2.1.1.1 RegistrationView

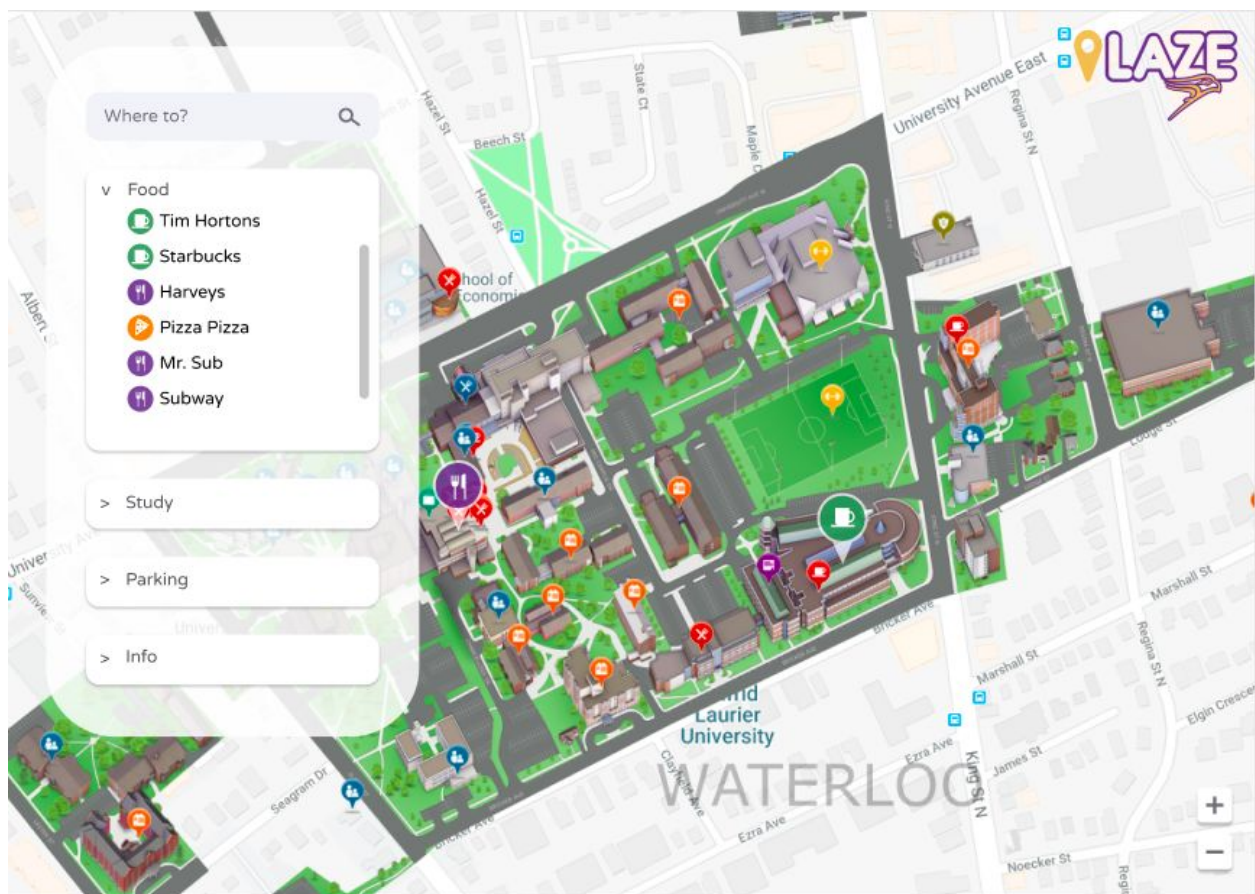
Field	Type	Description
username	input text	The user's desired name
username-availability	label	Dynamic label that displays if the entered username is available
email	Input text	The users confirmation email address
password	input text	The user's desired login password
confirm-password	input text	User confirms their desired password
password-check	label	Dynamic label that displays any errors regarding passwords not meeting requirements or if the two passwords don't match
tos-agreement	radio button	User checks this if he agrees to the terms of service
terms-of-service	button	The text will be a link that if clicked will link the user to the terms of service.
register	button	Registers the account and sends out confirmation email.

### 2.1.1.2 ForgotPasswordView

Field	Type	Description
email	input text	User enters the recovery email address

		registered when the account was created
send	button	Submits the email entered to receive the password-change email
confirmation	label	Dynamic label that tells the user if the email was sent successfully or if the email entered is not in the database.

## 2.1.2 MainView



Field	Type	Description
whereTo	input text	Enter keywords to search pins
food	drop down	Drop down menu that displays all current food related pins
study	drop down	Drop down menu that displays all current

		study related pins
parking	drop down	Drop down menu that displays all current parking related pins
info	drop down	Drop down menu that displays all current info related pins
zoomIn	button	Zooms the map in
zoomOut	button	Zooms the map out
mapPin	button	Any active pin on the map when clicked will bring up a dialogue containing the details of that pin.
addPin	button	Brings up the CreatePinView

### 2.1.2.1 CreatePinView

Field	Type	Description
mapselector	map	User clicks on the map where they would like to place their pin
category	drop down	Drop down menu that allows the user to select the category of their pin (ex. Food, study, parking, etc.)
details	input text	User enters the details of their pin
submit	button	Submits the user's pin to be posted
exit	button	X button that allows the user to cancel pin creation and return back to MainView

## 3. Detailed Design & Decomposition

### 3.1 Module Detailed Design

#### 3.1.1 Login View, RegistrationView, & ForgotPasswordView

Function Name	Parameters	Description
Login()	Email address, encrypted password, rememberMe (boolean)	Takes login information and allows or denies the user access to the next page
ResetPassword()	Email address	ResetPassword takes an email address, and if it matches an email address in the system, it sends a password reset link to that address
CreateUser()	Email address, name, username, encrypted password, profile photo (optional)	Creates a user in the database. Returns the result of action (i.e. it worked, or there was an error)
DeleteUser()	Email address, encrypted password	Deletes the user from the database. We must decide which user data will be orphaned and which data will be deleted.
SetUserData() (setters of the User class)	Email address, New user data	Functions relating to updating user data. The setters of the User class. Changes will be logged.

### 3.1.2 MainView (map page)

Name	Parameters	Description
Search()	Query (string)	Searches the various aspects of pins for relevant information. Returns a list of pins.
DeletePin()	Email address, pin id	Will verify then remove the pin and related data. Action will be logged.

### 3.1.3 Pins

#### 3.1.3.1 CreatePinView

Name	Parameters	Description
CreatePin()	Email address, Pin location, pin type, pin description, pin name/title	Verifies information and creates a pin. Notifies the front end of the result. Logs information about pin creation.
SetPinData() (setters of the Pin class)	Email address	Pin data will be updated with setters of the Pin class. Changes will be logged.

### 3.1.3.2 Pin class

#### 3.1.3.2.1 Pin class Properties

Modifiers	Type	Name	Description
private	int	ID	Uniquely identifies this pin
private	String	Description	Describes the purpose of and reason for the pin
Private final	Point2D	Location	Describes the location of the pin on an XY plane (potentially



			we might want to replace this with a Point3D)
Private final	EmailAddress	CreatedBy	The email address of the user who created this pin
private	int	VoteCount	The sum of all upvotes and downvotes of this pin. This number should always be updated whenever a user votes or changes their vote. This number should only be changed by the UpVote() and DownVote() function, and should not have a direct setter function.
Private	Collection<Vote>	VoterList	A collection of Vote objects which describes who has voted and how they voted
private	String	Name	The name (short description) of the pin
private	boolean	FlaggedAsBad	If someone has marked this pin as spam, this boolean should be set to true. Indicates that this pin might contain malicious information (like spam).
private	Collection<EmailAddress>	flaggers	A collection of the email addresses of the people who have flagged this post as bad
Private final	DateTime	dateTimeCreated	The date and time that this pin was created
Private final	DateTime	dateTimeToRemove	The date and time that this pin will expire. If NULL, this pin will never expire.

#### 3.1.3.2.2 Pin Functions

Name	Parameters	Description
------	------------	-------------

Pin()	Name, Description, Location, CreatedBy	Constructor of the Pin object
UpVotePin()	Email address	Increases the Rating of [Pin ID], and records the rating and voter to VoterList of the pin.
DownVotePin()	Email address	Decreases the Rating of [Pin ID], and records the rating and voter to VoterList of the pin.
FlagAsBad()	Email address, pin id, reason (enum), description	Allows users to flag pins as bad (spam, inappropriate, etc). Adds pin to a list of suggested actions for moderators to take.

### 3.1.3.3 BusinessPin class extends Pin

#### 3.1.3.3.1 BusinessPin Properties

Modifiers	Type	Name	Description
private	BusinessHours	openHours	Describes the times when the business is open. Sets dateTimeToRemove to null by default
private	int	congestion	Describes extent of how busy the business is
private	Icon	businessIcon	A special picture that makes people think of this business

### 3.1.3.4 Vote class

#### Vote Properties

Modifiers	Type	Name	Description
private	EmailAddress	voter	Identifies the user who created this vote
private	dateTime	createdAt	The date and time this vote was cast
private	boolean	isUpvote	Describes the direction of the vote. True if the vote is an upvote, False if the vote is a downvote.

### 3.1.3.5 EmailAddress class

#### Email address properties

Modifiers	Type	Name	Description
Private final	String	Local-part	The part before the @ symbol in the domain name
Private final	String	domain	The domain the email is at. Should be wlu.ca or mylaurier.ca
Private final	dateTime	createdAt	The date and time this vote was cast

#### Email address functions

Name	Parameters	Description
------	------------	-------------

EmailAddress	Local-part, domain	Constructor method which also verifies the information as described here ( <a href="https://en.wikipedia.org/wiki/Email_address#Syntax">https://en.wikipedia.org/wiki/Email_address#Syntax</a> ). Will limit to a set of Domains if requested.
--------------	--------------------	--

#### 3.1.3.6 BusinessHours class

BusinessHours properties

Note: the setters for this class must verify that the opening time for each day is before the closing time.

Modifiers	Type	Name	Description
private	array<time>[7]	openingTimes	The time the business opens each day of the week. Day 0 is Sunday, Day 6 is Saturday.
private	array<time>[7]	closingTimes	The time the business closes each day of the week. Day 0 is Sunday, Day 6 is Saturday.

#### 3.1.3.7 User

Modifiers	Type	Name	Description
Private final	EmailAddress	emailAddress	The email address of this user. This address is used as a unique identifier for this user.
private	String	name	The real name

			of the user
private	String	username	The displayed name of the user. Does not need to be unique
private	EncryptedPassword	password	The password of the user. This property is sensitive, and needs to be handled carefully, and securely.
private	Image	profilePic	The picture the user wants to display on their profile

#### 3.1.3.8 Moderator extends User

Being of the class Moderator gives the user some additional powers.

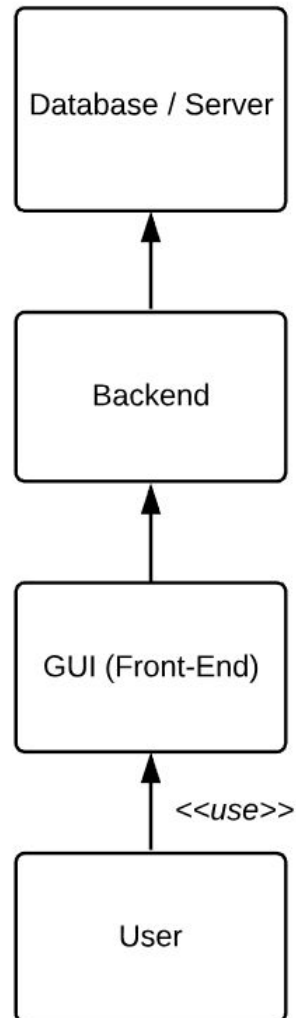
#### 3.1.3.8 Administrator extends Moderator

Being of the class administrator gives the user some additional powers.

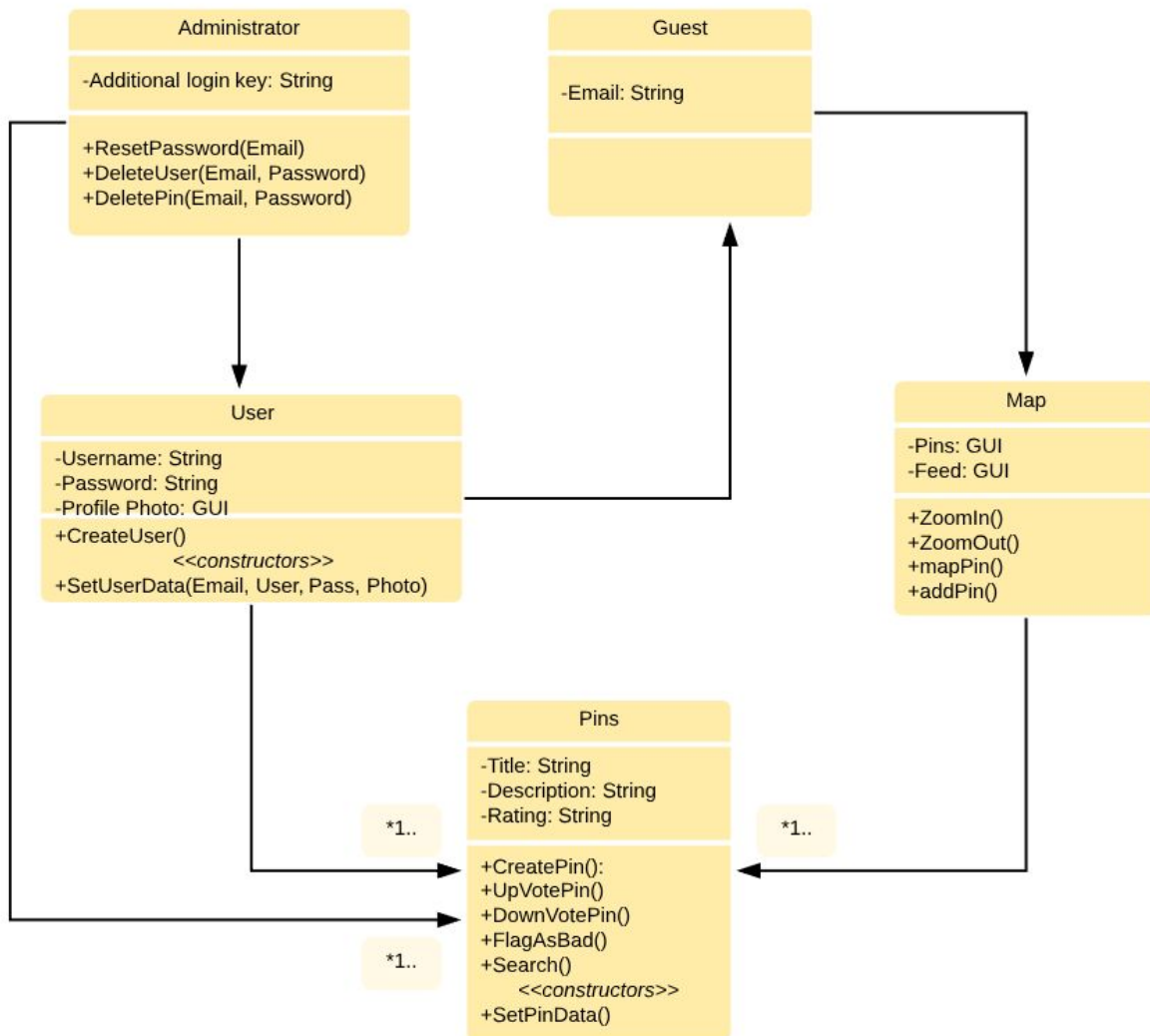
## 3.2 Module Decomposition

- Interaction diagrams
  - Framework Package Interactions
- Detailed class diagrams
  - Model Class Diagrams

## Laze Framework Package Interactions



## Laze Model Class Diagram



### 3.3 Coding Standards & Documentation

- Development tools
  - Virtual Environment
- Front end
  - HTML & CSS

- Bootstrap
  - JQuery
- Back End
  - Django==2.1.2 (libraries listed)
    - Celery
  - Python 3
- Hosting (TBD, options listed)
  - Google
  - Github pages
  - DigitalOcean
  - Python Anywhere
- Email System (TBD, options listed)
  - Mailchimp
  - Django

## 4. Versions

### Version 1.0 - 2018/11/04

*Created the document*

- Marek Szymczyk
- Mansi Bhatia
- Max Niebergall

### Version 1.1 - 2018/11/20

*Editing the document*

- Zizheng Huang