Laze

Table of Contents

Introduction

Purpose

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

Product Scope

The scope of this project is to assist in informing those around Wilfrid Laurier Campus with live updating information. Users will have the ability to input 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 - Sofware Engineering. Sofware Design Document for Electronic Scrolling Displays. TGR Karthik, 2006

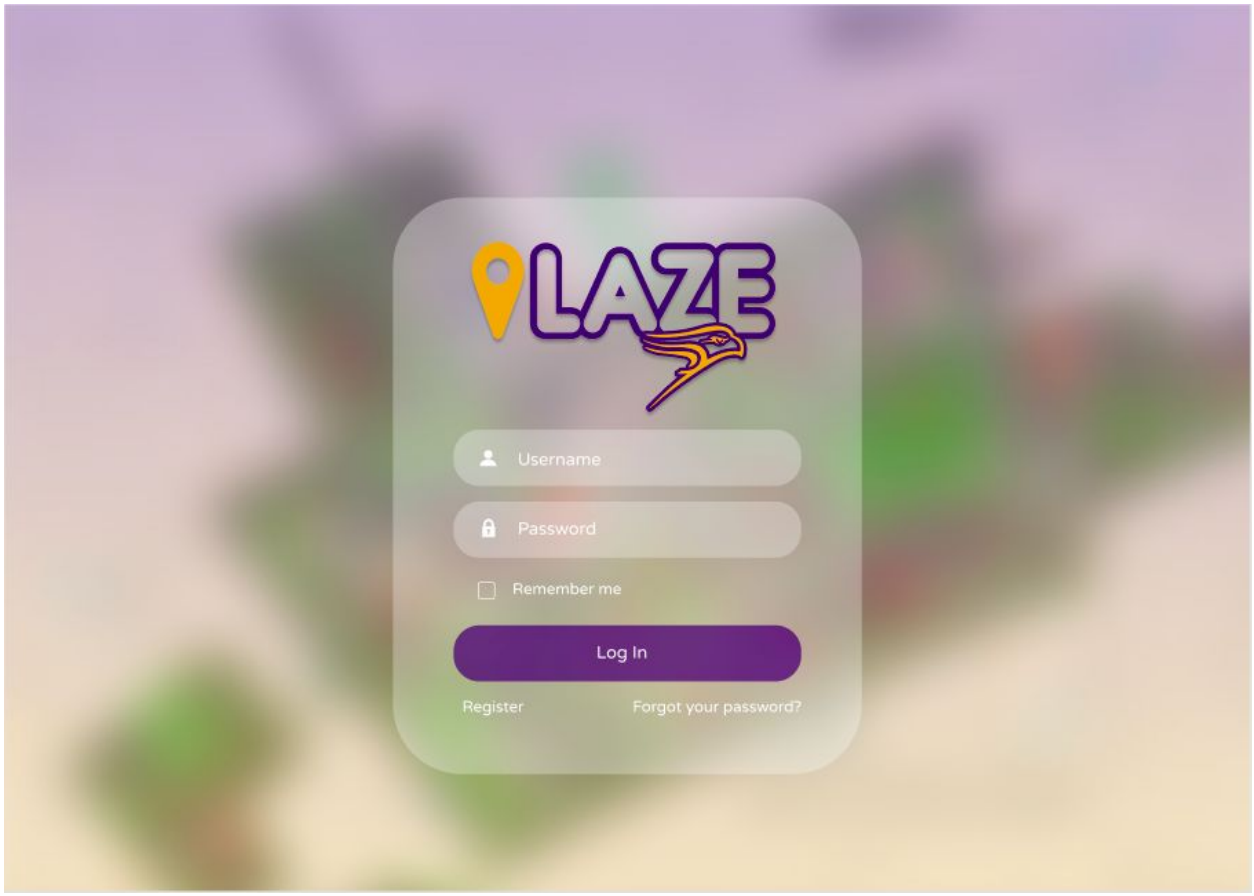
Overview

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

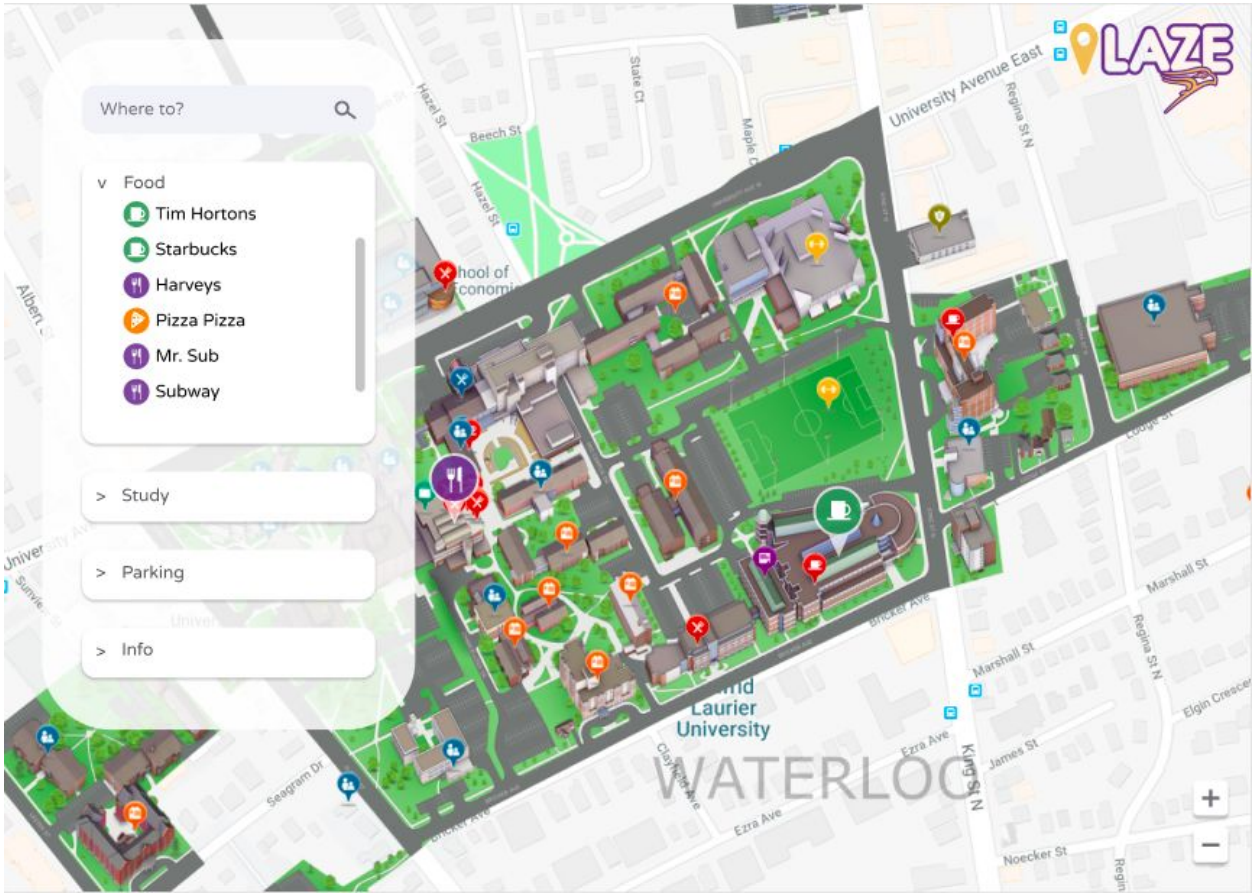
Field	**Type**	**Description**
username	input text	The user's desired name
username-availability	label	Dynamic label that displays if the entered username is available
email	Input text	The users confirmation email address
password	input text	The user's desired login password
confirm-password	input text	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 given

		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
info	drop down	Drop down menu that displays all current food 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

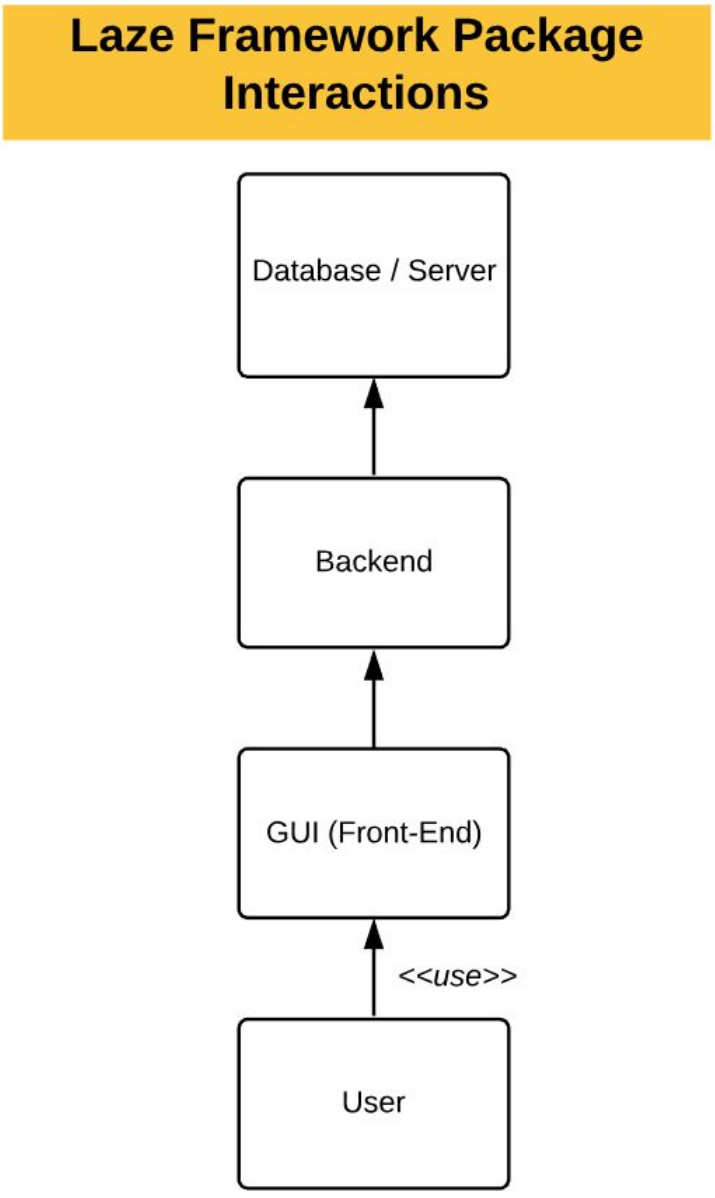
Function Name	Parameters	Description
Login()	Email address, encrypted password, rememberMe (boolean)	Takes login information and allows or denies the user access to the next page
ResetPassword()	Email address	ResetPassword takes an email address, and if it matches an email address in the system, it sends 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 will be setters of the User class. Changes will be logged.

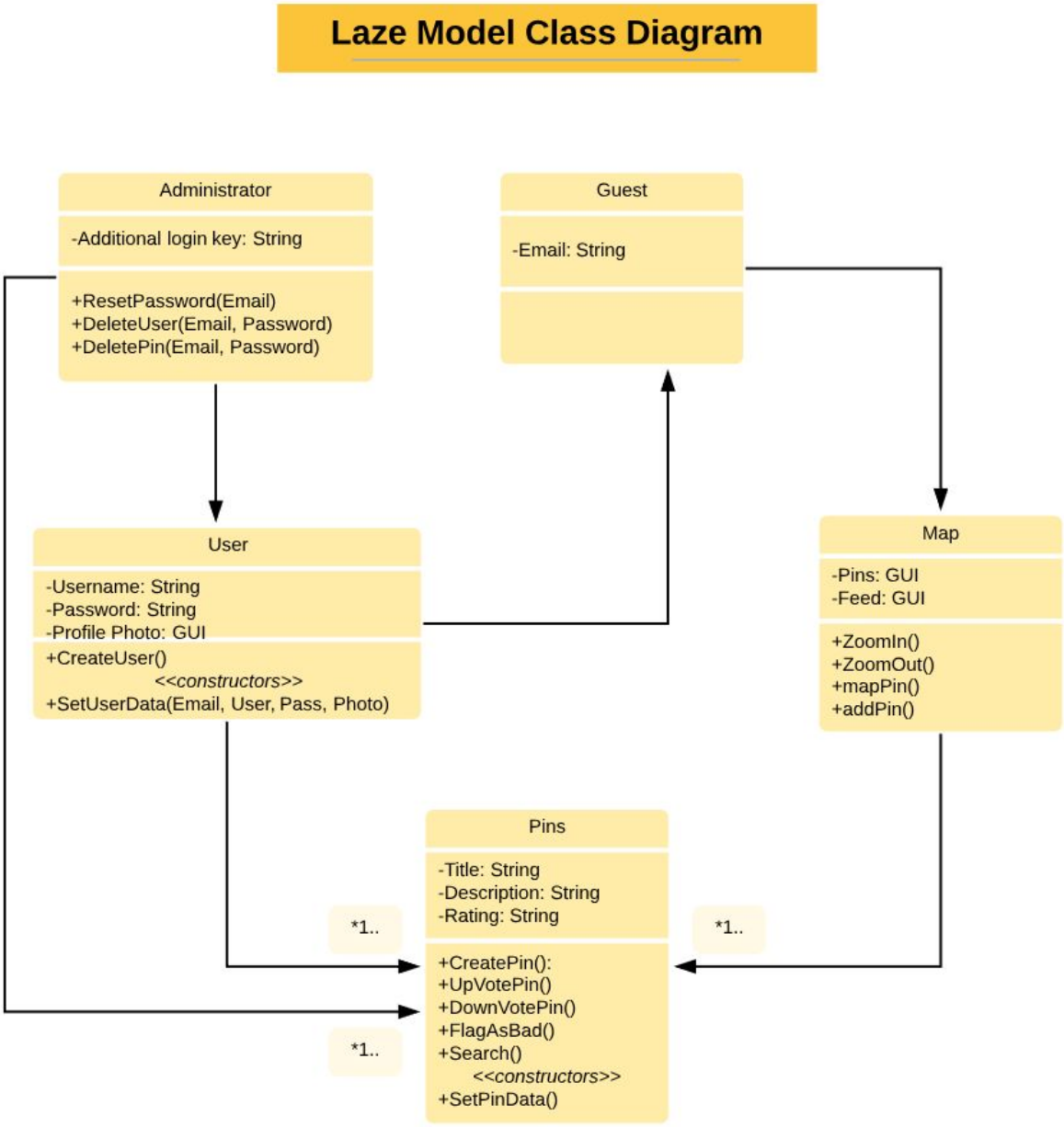
3.1.2 MainView (map page) and CreatePinView

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.
DeletePin()	Email address, pin id	Will remove the pin and related data. Action will be logged.
UpVotePin()	Email address, pin id	Increases the Rating of [Pin ID], and records the rating in the list of users which have rated this pin.
DownVotePin()	Email address, pin id	Decreases the Rating of [Pin ID], and records the rating in the list of users which have rated this 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.
Search()	Query (string)	Searches the various aspects of pins for relevant information. Returns a list of pins.

3.2 Module Decomposition

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





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