



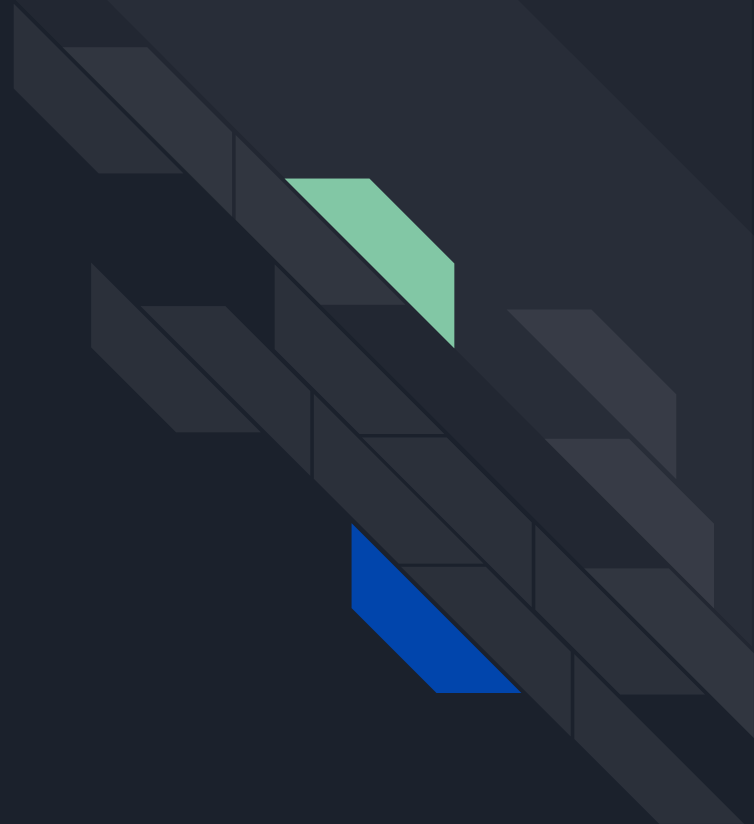
# Plan-it

On the go spontaneous travelling

A project by ...  
William Granados, Jack Woodger, Anh Le,  
Lucy Tishkina, Daniel Zhao

# Product Introduction

*Phase I*



# Travelling Apps and Technology



TripAdvisor



Expedia



Yelp



Google Maps

# TripAdvisor



Toronto ▾



JOIN

Search

Toronto Hotels **Things to do** Restaurants Flights Vacation Rentals Shopping Package Holidays Rental Cars

Canada > Ontario > Toronto > Things to do in Toronto > Activities & Games in Toronto

Top Fun Activities & Games in Toronto, Ontario

## Fun Activities & Games in Toronto

When are you travelling?

Start Date

End Date

Search

< Back to All Things to Do

View Map

< All Things to Do

Your Selections

Clear all

\* Fun & Games

Fun & Games

Game & Entertainment Centres (64)

Room Escape Games (38)

Movie Theatres (26)

### Sports Complexes (13)

See all



Rogers Centre

3,892...

Arenas & Stadium



Trinity Bellwood...

107 R...

Sports Complexes



Lampport Stadium

23 Re...

Arenas & Stadium



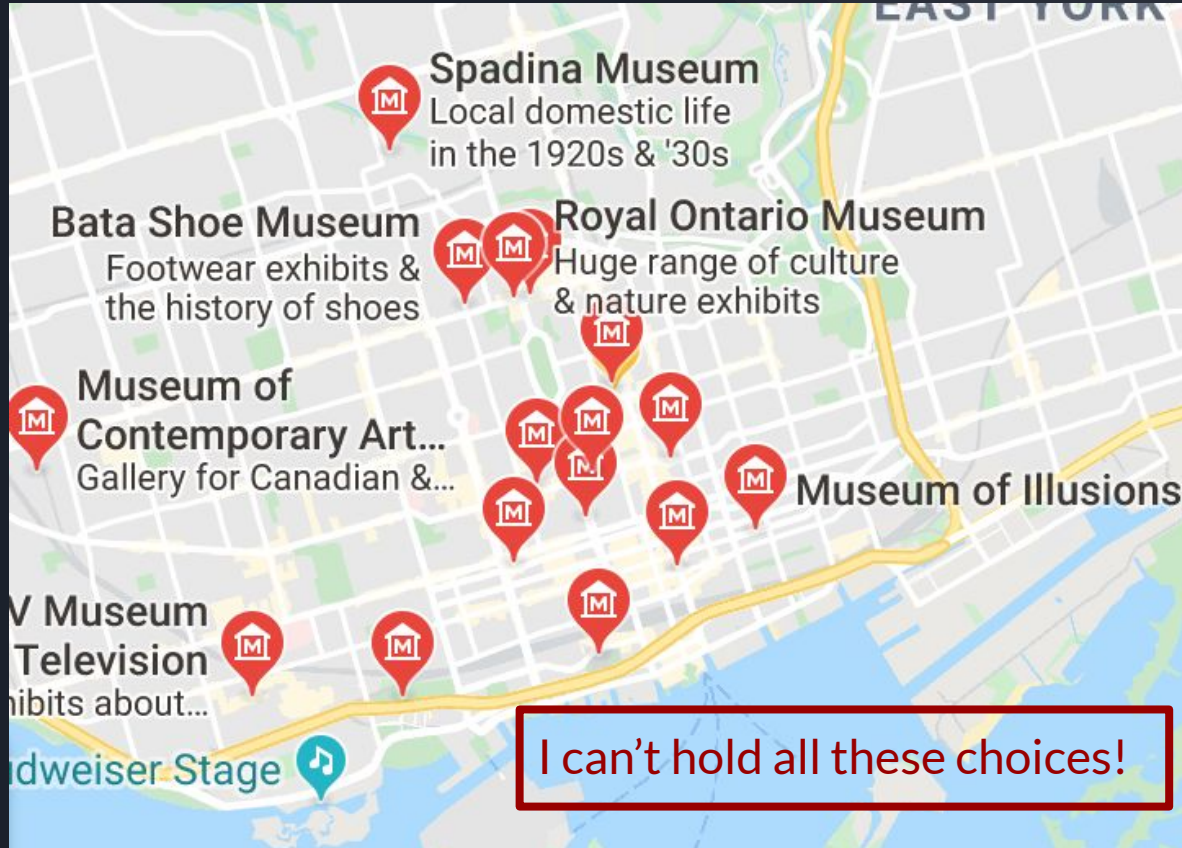
True North Clim...

12 Re...

Sports Complexes

Tons of options, but user needs to choose

# Google Maps



# Purpose of Plan-it Save the **planning**



# Purpose of Plan-it

The missing niche: plan-it for me

The background features a series of dark grey, 3D-style rectangular blocks arranged in a descending staircase pattern from the top right towards the bottom left. Two specific blocks are highlighted: a light green one and a blue one, both positioned in the middle-right section of the staircase.

# Let's Plan-it!



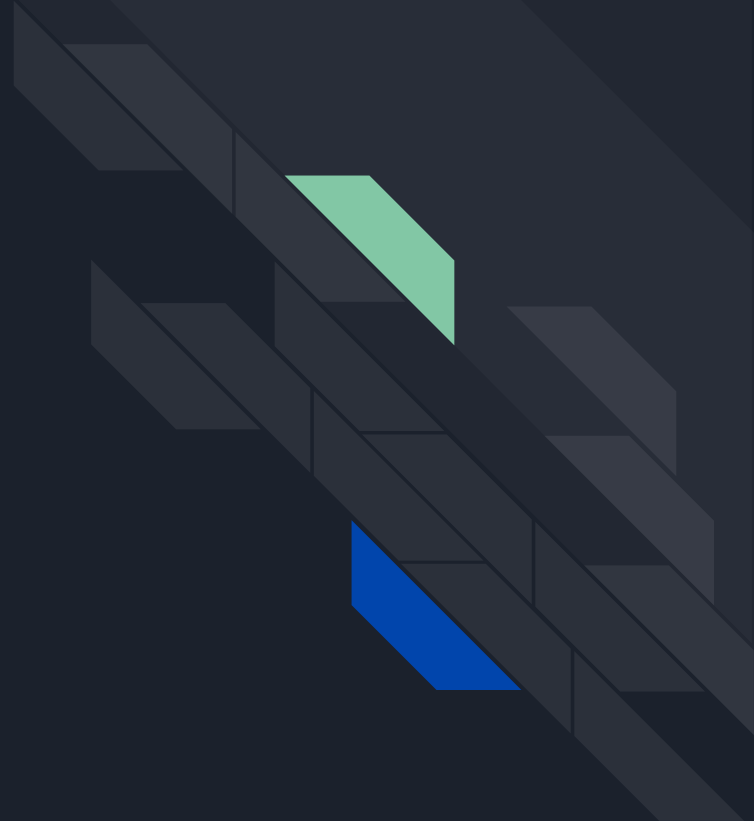


# What Makes Plan-it Unique

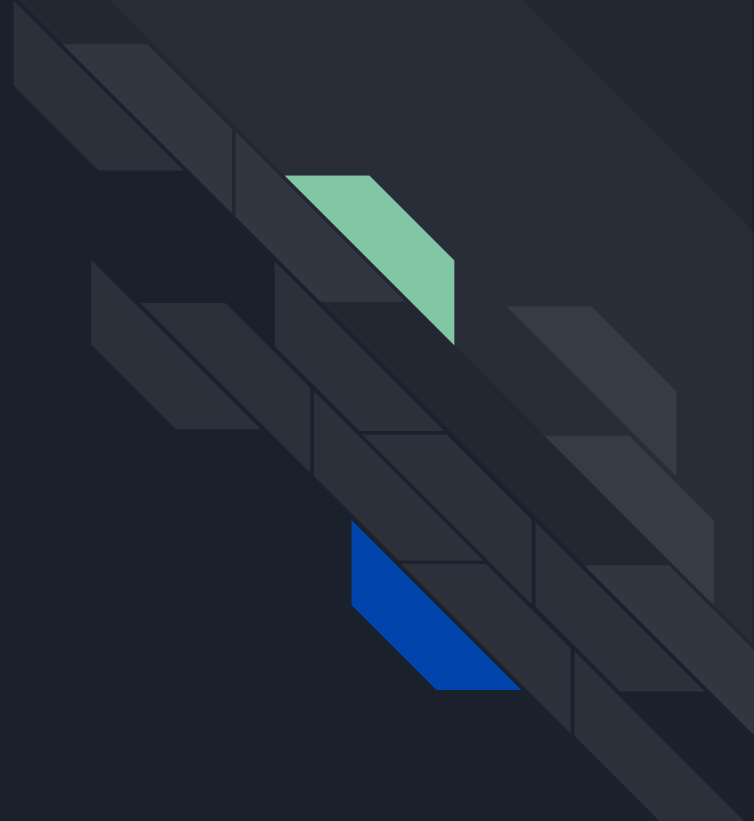
- 1) Random Venue Generation
- 2) Trip History (Personalization System)
- 3) Filters(Travel Modes), with Google Maps Integration

# Process

*Phase II*



# The Team



# Distribution of work

Lucy – worked on map, and routing for destinations, api for venues

Daniel – search filters, CSS, display of search results through api

Jack – Itineraries & trip history

Anh – SQL database management

William – login & reviews, mongodb

# Challenges of Development Process

- front end java frameworks are challenging
  - JSP & thymeleaf very different, and incompatible, and several tutorials for both
  - addressed this through meetings and deciding best framework our needs
- Merging different features with dependencies
  - Itinerary and Trips and Login were isolated for a large portion of the sprint, and when we needed to merge there were conflicts
  - addressed this through feature branches and GitFlow
- database management, integrations, and consistency
  - database consistency across our laptops
  - unfamiliarity with database technology (MYSQL, NOSQL) with MVC
  - works on my laptop but not yours!

# Highlights of development process



Pull requests  
Include:  
images,  
description,  
code reviews!

UTSCSCC01 / [project-not-tripadvisor](#) Private

Unwatch 4 Star 0 Fork 0

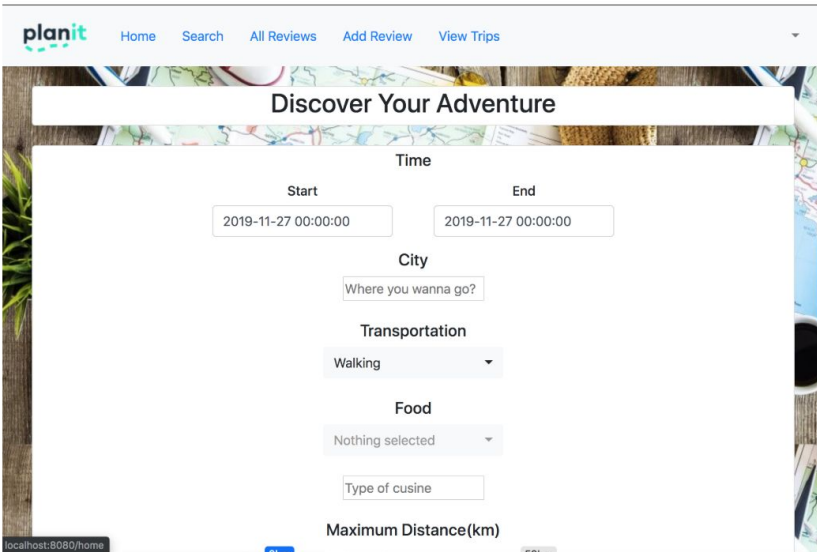
Code Issues 0 Pull requests 1 Actions Projects 0 Wiki Security Insights

## Beautify #19

Open codedcomposer wants to merge 4 commits into master from beautify

Conversation 7 Commits 4 Checks 0 Files changed 27 +699 -279

codedcomposer commented 4 days ago • edited



The screenshot shows a web application interface for planning an adventure. It features a navigation bar with 'Home', 'Search', 'All Reviews', 'Add Review', and 'View Trips'. The main content area is titled 'Discover Your Adventure' and contains several form fields: 'Time' with 'Start' and 'End' date pickers (both set to 2019-11-27 00:00:00), 'City' with a text input 'Where you wanna go?', 'Transportation' with a dropdown menu set to 'Walking', 'Food' with a dropdown menu set to 'Nothing selected', and a 'Type of cuisine' text input. At the bottom, there is a 'Maximum Distance(km)' label. The browser's address bar shows 'localhost:8080/home'.

Reviewers: wlganados

Assignees: No one—assign yourself

Labels: None yet

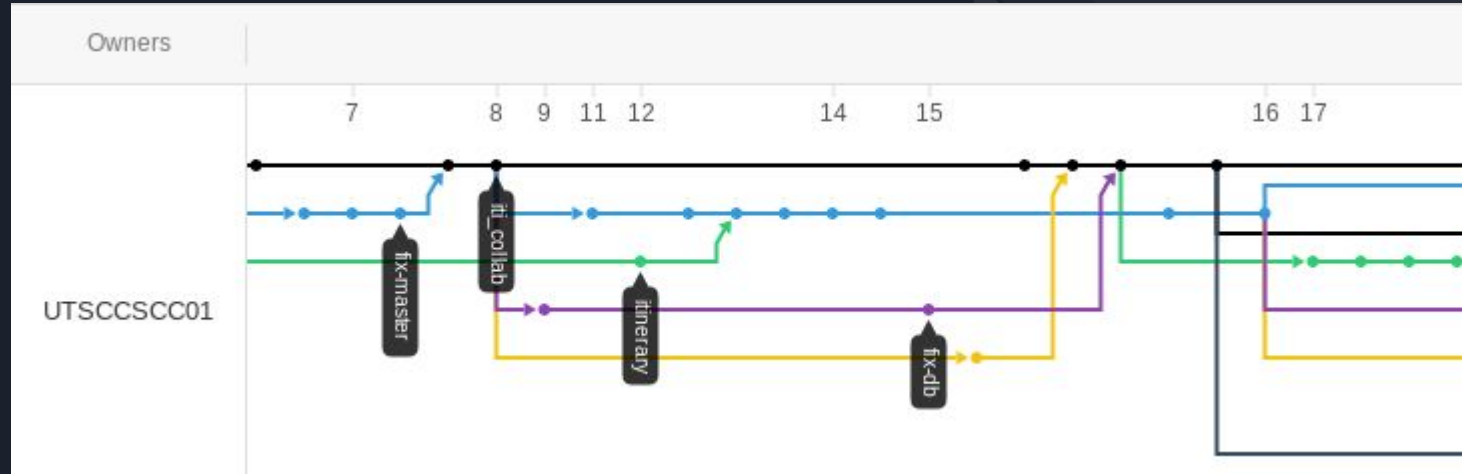
Projects: None yet

Milestone: No milestone

Notifications: Unsubscribe

You're receiving notifications because you're watching this repository.

# Git workflow



No committing to master!  
Only feature branches!

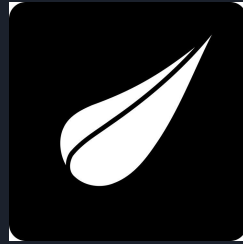


# Scrum Process

Product Backlog  
Sprint Backlog  
Daily Stand Up  
Planning Poker  
Sprint Reviews

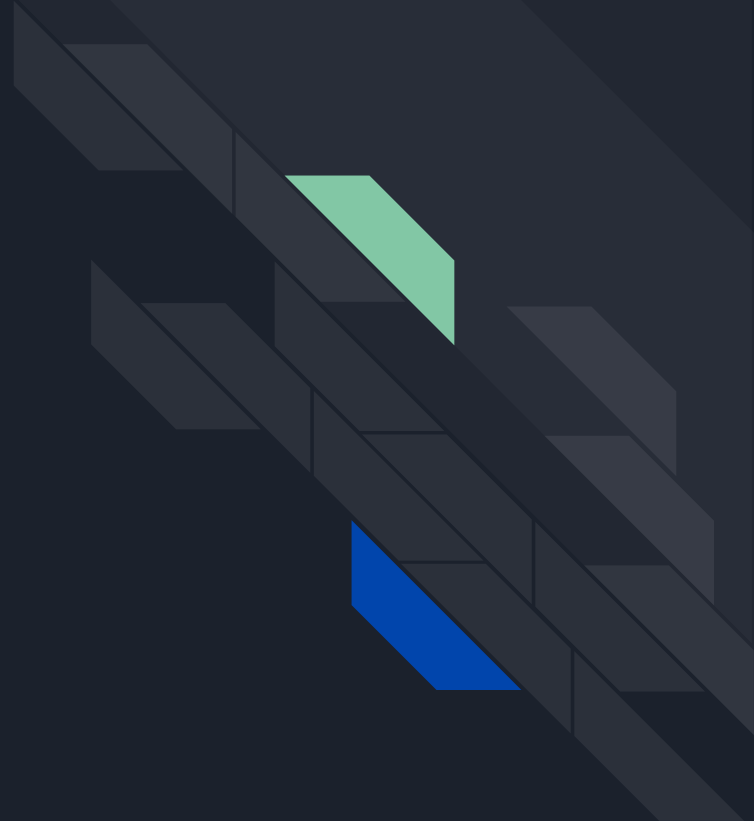


# Tools & Technologies

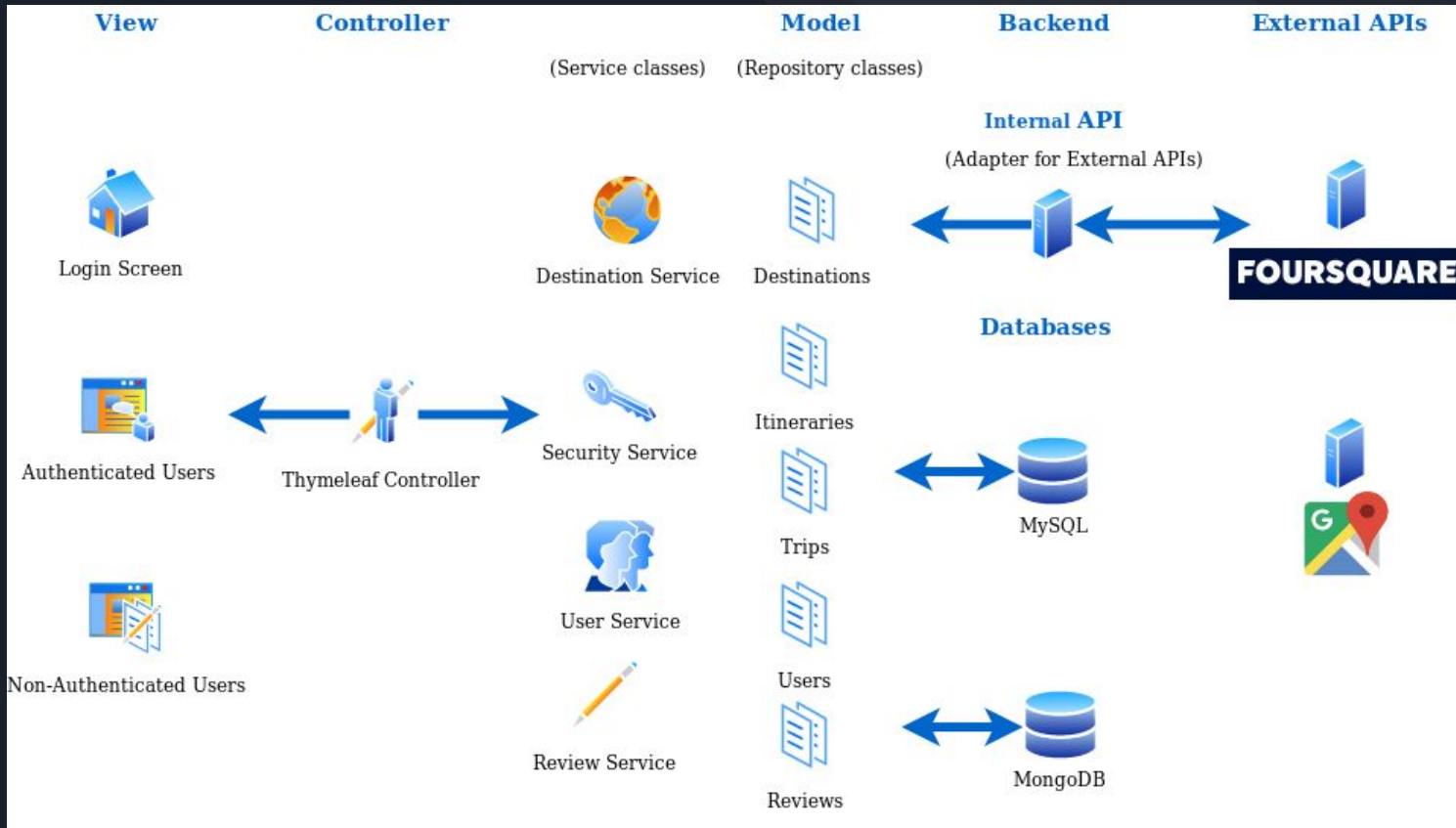


# Software Architecture

*Phase III*

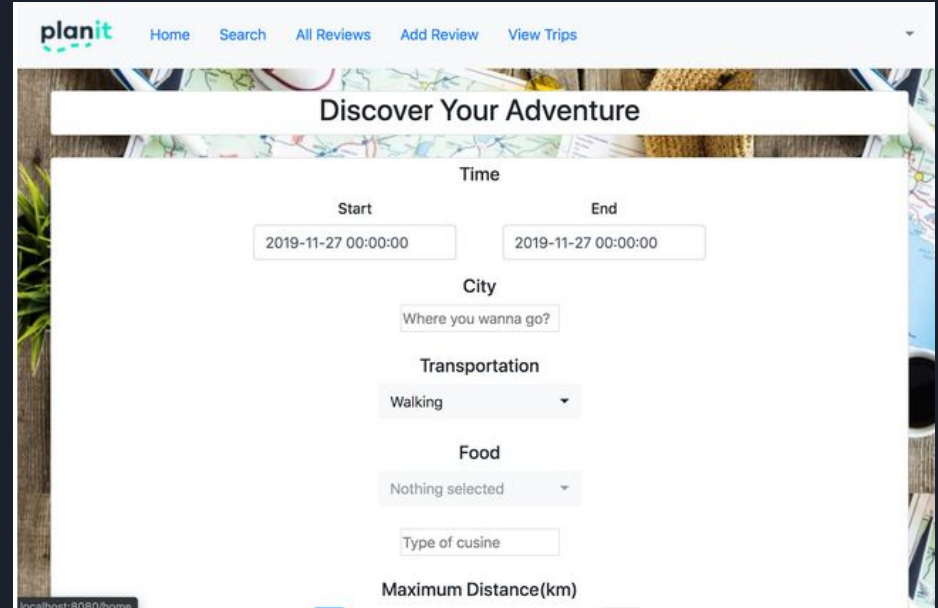


# Software Architecture



# View: Frontend

- Displays content to user based on user's role/status
- User experience/app flow different for users who are registered and logged allowing access to additional features
- Displays model data to user

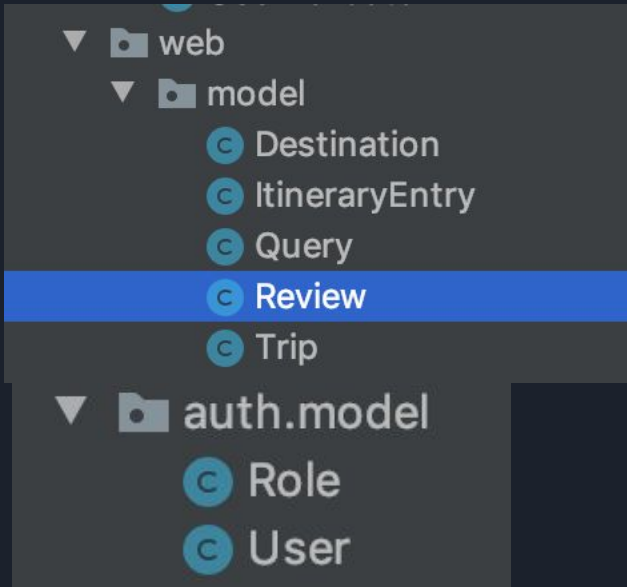


The screenshot shows the 'Discover Your Adventure' form on the planit website. The form is set against a background image of a map and travel items. At the top, there is a navigation bar with the 'planit' logo and links for 'Home', 'Search', 'All Reviews', 'Add Review', and 'View Trips'. The form itself is titled 'Discover Your Adventure' and contains several input fields and dropdown menus:

- Time:** Two date-time input fields labeled 'Start' and 'End', both containing the value '2019-11-27 00:00:00'.
- City:** A text input field with the placeholder text 'Where you wanna go?'.
- Transportation:** A dropdown menu currently showing 'Walking'.
- Food:** A dropdown menu currently showing 'Nothing selected'.
- Food:** A text input field with the placeholder text 'Type of cuisine'.
- Maximum Distance(km):** A label for a field that is partially visible at the bottom of the form.

At the bottom left of the browser window, the address bar shows 'localhost:8080/home'.

# Model



Exposes and stores the application data

Keeps track of applications state

Responds to queries and updates





# Backend

## External apis

- We used Foursquare, for destinations and itineraries, and points of interests

## Mongodb

- We used for storing review information

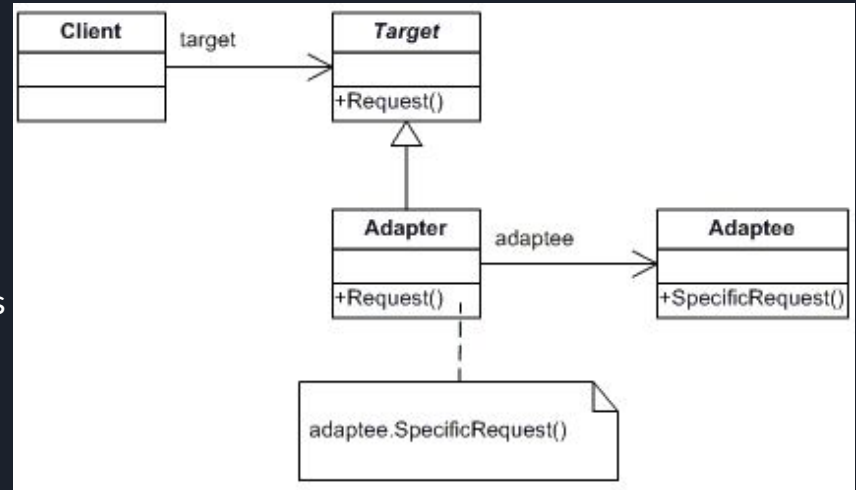
## Mysql

- We used for storing sensitive user information



# Design Patterns: Adapter

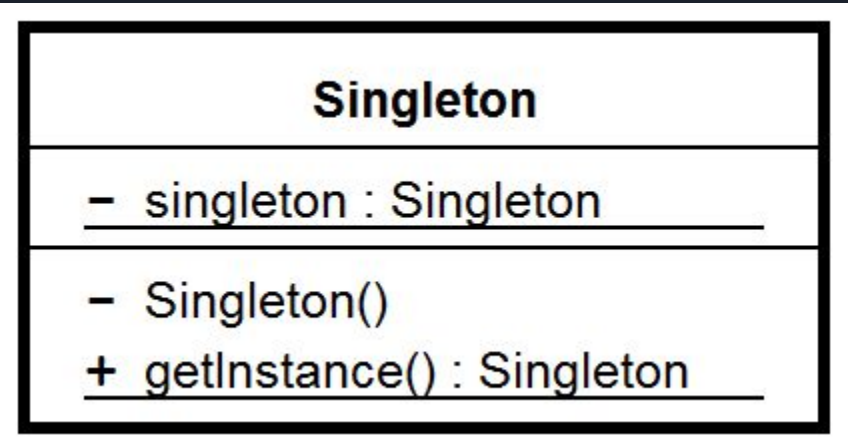
- Internal API acts as an adapter between the external API's
- The queries required for our application are converted by the backend API to queries to FOURSQUARE API to fetch the relevant content
- For example, the internal API aggregates results for points of interest indicated by the user and returns the resulting query by calling the external APIs





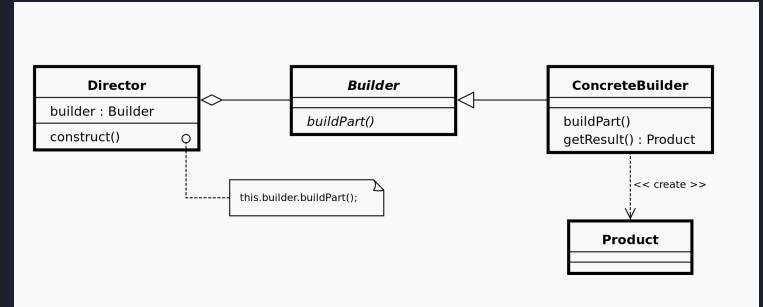
# Design Patterns: Singleton

- Database connections use singleton



# Design Patterns: Builder

- Used for the construction of Destination objects
- Destinations can have a subset of fields instantiated at initialization, which would require many different constructors



# Access Control

**Problem:** How do we restrict access to features for Users

**Solution:** Java spring security

- Requiring User Objects to be present when rendering certain views
- Integration with SQL to encrypt the passwords and check hashes for authentication
- Storing information about the user in the session as a user object

