Research Practicum Project Report

Predicting Dublin Bus Journey Times

Ratnaprabha Keshwar (19200371)

A thesis submitted in part fulfilment of the degree of

MSc. in Computer Science (Conversion)

Group Number: 17

COMP 47360

UCD School of Computer Science

University College Dublin

August 18, 2020

Project Specification

Actual journey time taken by city transport buses usually differs from published time schedules due to dynamic variables like Traffic conditions, Time of day, Day of the week, Month of the year, Weather, etc. This makes planning the trips on public transport modes a bit difficult. This project involves analyzing historic Dublin Bus data and Dublin weather data to create dynamic travel time estimates.

The system should be a user interactive, web-based interface optimized for handheld devices. It should enable users to choose a bus route, an origin stop, a destination stop, and date-time to begin the journey. Based on these inputs along with weather conditions, the system should generate and display estimated travel time for the selected journey and sections of the route.

Abstract

Dublin Bus operates bus service in Dublin city along with surrounding counties of Meath, Wicklow, and Kildare. Huge passenger bases of such systems are interested in knowing accurate bus schedules as well as projected journey times of later days. This project report discusses the approach behind the development of a "Dublin Bus web application" addressing these issues. It also describes a methodology used to deliver an intuitive map-based application, which provides accurate predictions for bus arrival and total journey times. Unification of complementary features like leap card balance, cost of the journey, and weather at the time of the journey is an added advantage for users of the application.

Acknowledgments

The completion of this project required a lot of guidance and we were fortunate to receive it throughout the project by our guides and mentors. We take this opportunity to humbly thank Dr. Gavin McArdle for supporting us in the timely completion of the project. We are also grateful to him for his insights which helped us in delivering this end to end web application. We would also like to express our gratitude to our project mentors Dr. Soumyabrata Dev, Dr. Félix Balado, and Dr. Pádraig Cunningham for their strict and honest feedback which motivated us to improve the deliverables.

This project was a team effort driven by our group members - Sarvesh Gadgil, Rasik Kane, Ratnaprabha Keshwar and Swetha Bondhi Krishnamurthy. Lastly, we would like to thank UCD School of Computer Science for this collaborative learning opportunity.

Table of Contents

Chapter 1: **Introduction**

Dublin Bus provides services to Dublin city and its surrounding counties. Dublin Bus employs over 1,010 vehicles running on more than 120 routes, covering over 57.1 million kilometres annually. It operates a fleet which primarily consists of double-decker buses, carrying over 143 million passengers every year [1].

Prime requirements of regular passengers are knowing accurate bus schedules and projected journey times. For an educational and industrial hub like Dublin; traffic density and weather have a considerable impact on urban transport. A unified web application addressing these issues could reduce long queues and waiting times for city commuters.

This project delivers a working web application that assists Dublin Bus users to plan their journey. The web application enables users to foresee time for journeys, along with route visualization on the map. Journey time weather, leap card balance, and cost of the journey are also included.

1.1 Problem Statement

Dublin Bus schedules are generic and statically defined. Whereas, weather and varying density of traffic pertaining to the area and time of day are dynamic conditions. Hence, defining fixed arrival time for a bus to reach a stop would not be precise. Dublin Bus web application addresses this challenge. It provides an estimate of arrival time at each bus stop and total journey time. These estimates are predicted by machine learning models that are trained using historic Dublin Bus data.

1.2 Existing Approaches and Limitations

Multiple mobiles and web applications provide real-time information about Dublin Bus arrival times at particular bus stops [2]. These services are based on a fixed Dublin Bus timetable. Most applications rely on the RTPI API for real-time information. On the other hand, map-based solutions like hitting the road, Google Maps, etc. have a good user base and can help users to plan a multistage journey [3].

But with existing solutions; users are still unable to check bus schedules and journey times for later dates. Also, while commuting; users are concerned about weather conditions, leap card balance [4], incurred fare, etc. These factors are not integrated into any of the web-based solutions currently available to Dublin Bus users.

1.3 Overview

The website consists of a single page application with an autocomplete box and the map placed adjacent to each other. This layout provides a bird-eye view to the user. It lists available buses at bus stops along with a weather forecast at the time of the journey. Route of the journey is displayed on the map along with markers informing about the bus stop name, number, and arriving buses. Bus stops are color-coded; where black marker indicates onboarding stop, red markers represent on-route stops, and deboarding stops are shown with a green marker.

Journey time estimates including total travel time, arrival time at each bus stop, travel time between successive bus stops, and time to arrive at onboarding bus stops are also displayed.

Users can also check their Leap card balance. Leap card balance which would remain after taking a searched journey is also displayed below predicted journey time. Registered users are provided one-click predictions for their recently searched routes. Twitter feed from the Dublin Bus is also provided to registered users in the top right section.

Link to Website:

- Dublin Bus Application

Link to GitHub Repositories:

- Front End Code Base
- Back End Code Base

1.4 Structure of the Report

The report is organised into 5 chapters:

- Chapter 1 Introduction: This chapter discussed the problem statement and existing solutions along with an overview of delivered web application.
- Chapter 2 Description of Final Product: Necessary and additional features of Dublin Bus web application are briefly discussed.
- Chapter 3 Development Approach: Tools and processes used to implement AGILE Software Development Life Cycle and team management are discussed.
- Chapter 4 Technical Approach : Perspective behind system architecture and choice of technology stack is briefed.
- Chapter 5 Testing and Evaluation: Strategy for testing and bug resolution are detailed. Results of web environment testing and prediction model validation are discussed.

Chapter 2: **Description of Final Product**

2.1 Functionality of the Application

Dublin Bus web application is designed to provide accurate predictions of journey duration as well as the arrival time of the bus at all stops. Since urban commuters are the target users for application, the whole project is designed with a minimalistic interface. Application has a responsive user interface; providing better user experience on mobile devices.

2.1.1 Core Features

To implement an operational website for Dublin Bus travel time prediction; following features were implemented:

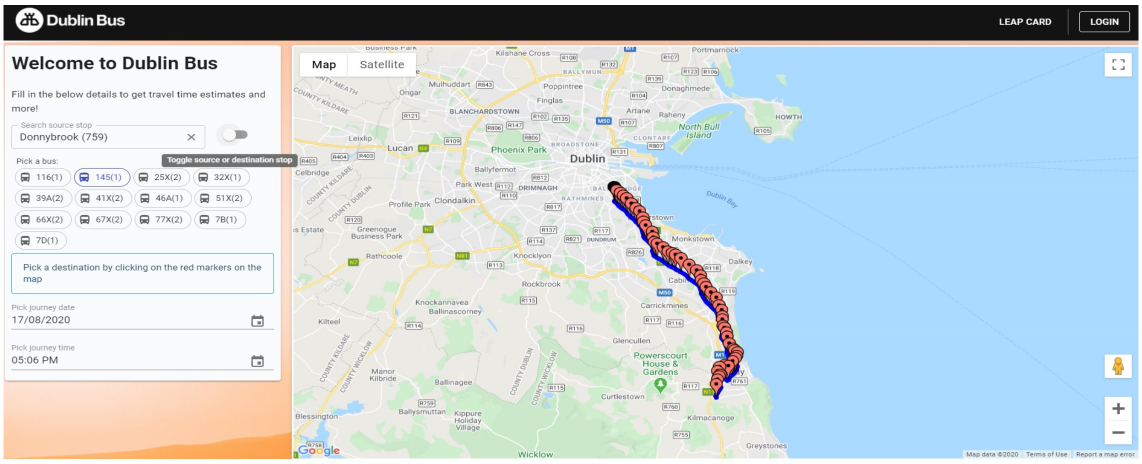
- Enable user to select date and time for journey.
- Provide a toggle button to search a bus stop as either source or destination.
- Suggestion for bus stops and dublin localities; as user types in search box.
- Displaying multiple nearby bus stops within proximity of selected dublin locality on embedded Google Map.
- Show buses served at selected bus stop and display route for a chosen bus on a map.
- Show predicted travel time for entire journey and for each section of the route.

2.1.2 Innovative Features

- Provide user registration with secure login/ logout facility.
- Displaying predicted temperature, feels like temperature and weather conditions at the time of the journey.
- Show predicted "time to arrive" at an onboarding bus stop.
- Display one-click access to bus timing and weather predictions for recent routes.
- Show estimated fare for a journey.
- Enable users to view Leap card balance, status, and expiry dates, etc.
- Display projected remaining balance if users take a searched journey.

2.2 Addressing Project Specifications

When users access the website, the side panel and map are visible. As the user starts typing the locality name using side panel, matching bus stop names from the database are shown. Along with that, Google powered Autocomplete shows relevant locality names matching entered text. Users can directly select one of the bus stops from the list of bus stops. Or, When a locality is chosen, all available Dublin Bus stops within a 0.5 km radius are shown on the map. Users can click and select the bus stop. All Dublin Buses served at the stop are displayed. The direction of the bus is also displayed, where "1" indicates inbound while "2" indicates outbound buses. When users select a bus and the toggle switch is on "Source", then the travel path from the selected source bus stop till the last bus stop is shown. Conversely, When the toggle switch is on "Destination", the travel path from the first bus stop to the selected destination bus stop is shown. All stops on the route are also shown.



Welcome to Dublin Bus

Fill in the below details to get travel time estimates and more!

Search source stop: Donnybrook (759)

Pick a bus:
116(1) 145(1) 25X(2) 32X(1)
39A(2) 41X(2) 46A(1) 51X(2)
66X(2) 67X(2) 77X(2) 7B(1)
7D(1)

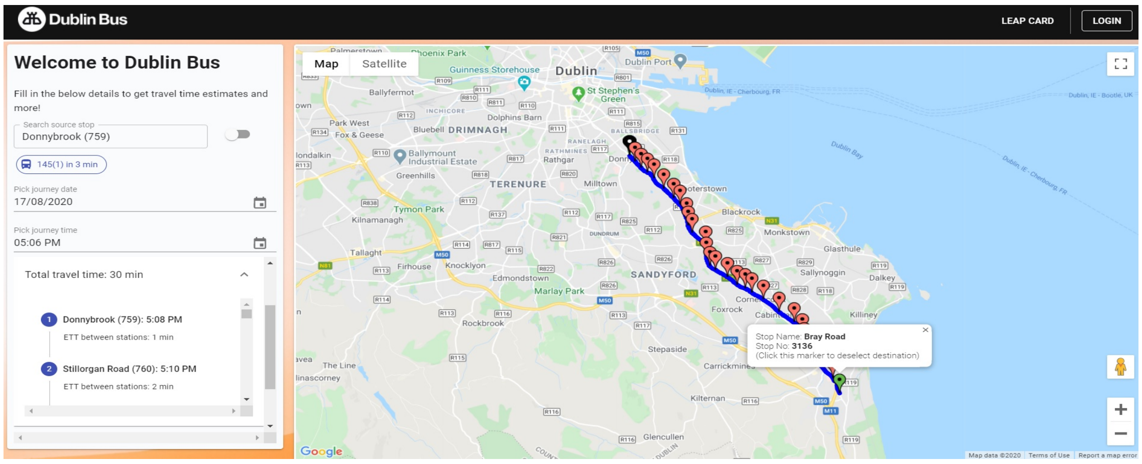
Pick a destination by clicking on the red markers on the map

Pick journey date: 17/08/2020

Pick journey time: 05:06 PM

Figure 2.1: Website: Searched Stop and Route

On the displayed route, users can select a bus stop marker. This selected bus stop is treated either as a boarding bus stop or as a deboarding bus stop; depending upon the toggle switch. Users may deselect the chosen bus stop marker to revert selected buses and change selected buses. As users now have selected a bus, source, and destination; total journey time predicted by the ML model is displayed using current date-time. Along with; vertical stepper shows when the bus arrives at each stop and expected travel time between consecutive stops. Users can also choose the date and time of the journey to view predictions over the next 4 days.



Welcome to Dublin Bus

Fill in the below details to get travel time estimates and more!

Search source stop: Donnybrook (759)

145(1) in 3 min

Pick journey date: 17/08/2020

Pick journey time: 05:06 PM

Total travel time: 30 min

1 Donnybrook (759): 5:08 PM
ETT between stations: 1 min

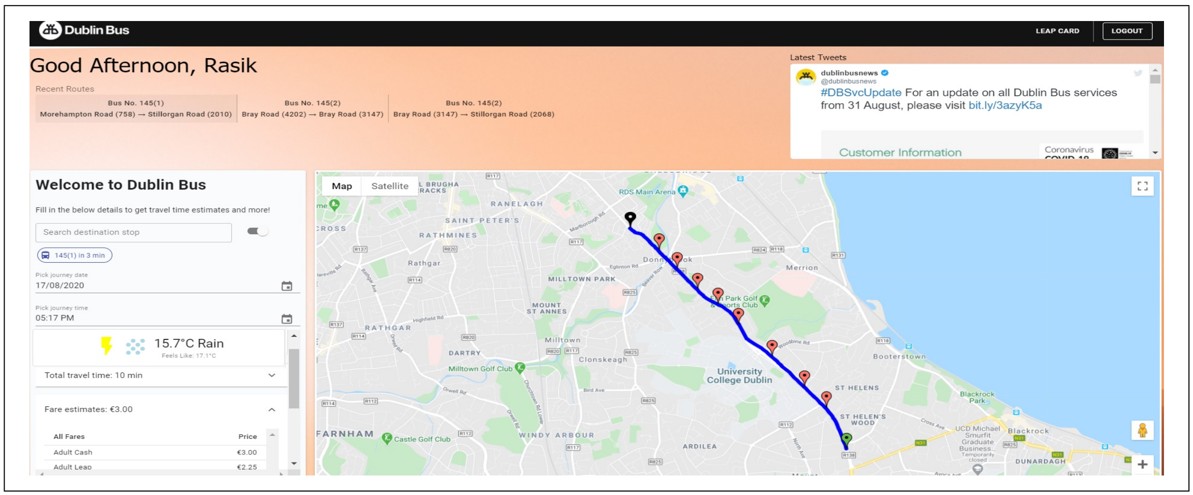
2 Stillorgan Road (760): 5:10 PM
ETT between stations: 2 min

Stop Name: Bray Road
Stop No: 3136
(Click this marker to deselect destination)

Figure 2.2: Website: After Travel Time Prediction

2.3 Innovative Solutions

Registration of a new user requires 4 mandatory fields viz. First name, Last name, Email ID, Password. Validation of fields is implemented. Registered users can log in using email ID and password as login credentials. Logged in users are presented with a greeting message along with their recently searched journeys and live twitter feed of Dublin Bus. Using recent routes, users can get one-click travel time predictions, weather predictions, fare estimates, and route visualization on the map.



Good Afternoon, Rasik

Recent Routes

Bus No. 145(1) Morehampton Road (758) → Stillorgan Road (2010)
Bus No. 145(2) Bray Road (4202) → Bray Road (3147)
Bus No. 145(2) Bray Road (3147) → Stillorgan Road (2068)

Latest Tweets

dublinbusnews @dublinbusnews
#DBSvcUpdate For an update on all Dublin Bus services from 31 August, please visit bit.ly/3azyK5a

Welcome to Dublin Bus

Fill in the below details to get travel time estimates and more!

Search destination stop

145(1) in 3 min

Pick journey date: 17/08/2020

Pick journey time: 05:17 PM

15.7°C Rain
Feels Like: 17.1°C

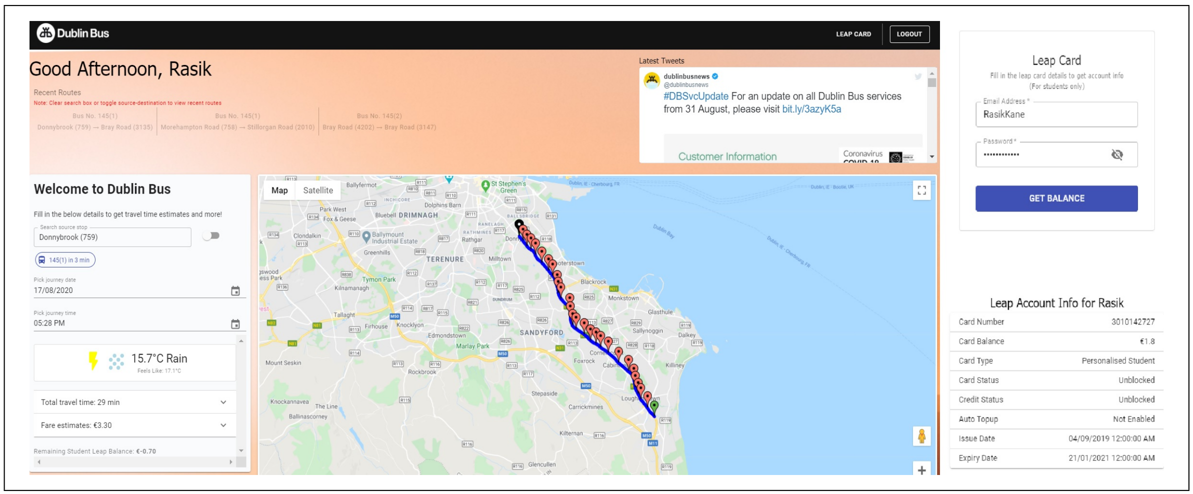
Total travel time: 10 min

Fare estimates: €3.00

All Fares	Price
Adult Cash	€3.00
Adult Leap	€2.25

Figure 2.3: Website: Weather,Twitter and Recent Routes

For all users, the expected arrival duration for the selected bus at the onboarding bus stop is displayed. This time is expressed in minutes or hours and is important in the case of buses with sparse timetables. Using number of stops travelled in a journey, possible fare for each passenger class [adult/ student/ children] in each payment form [cash/ leap] is displayed [5]. Users can also access their leap card overview. It lists card balance and status, date of expiry and card type, etc. Leap card balance which would remain after taking a searched journey is also displayed below predicted journey time.



Good Afternoon, Rasik

Recent Routes

Note: Clear search box or toggle source-destination to view recent routes

Bus No. 145(1) Donnybrook (759) → Bray Road (3135)
Bus No. 145(1) Morehampton Road (758) → Stillorgan Road (2010)
Bus No. 145(2) Bray Road (4202) → Bray Road (3147)

Latest Tweets

dublinbusnews @dublinbusnews
#DBSvcUpdate For an update on all Dublin Bus services from 31 August, please visit bit.ly/3azyK5a

Welcome to Dublin Bus

Fill in the below details to get travel time estimates and more!

Search source stop: Donnybrook (759)

145(1) in 3 min

Pick journey date: 17/08/2020

Pick journey time: 05:28 PM

15.7°C Rain
Feels Like: 17.1°C

Total travel time: 29 min

Fare estimates: €3.30

Remaining Student Leap Balance: €-0.70

Leap Card

Fill in the leap card details to get account info
(For students only)

Email Address *: RasikKane

Password *: ••••••••••••

Leap Account Info for Rasik

Card Number	3010142727
Card Balance	€1.8
Card Type	Personalised Student
Card Status	Unblocked
Credit Status	Unblocked
Auto Topup	Not Enabled
Issue Date	04/09/2019 12:00:00 AM
Expiry Date	21/01/2021 12:00:00 AM

Figure 2.4: Website: Leap Card and Remaining Balance

Chapter 3: **Development Approach**

3.1 Analysis of Resources

3.1.1 Problem Analysis

Project deliverables required a working web application providing total journey time prediction and travel time for subsections of the route. Hence, the team discussed and finalized the following components of the project:

- Frontend with a map for visualisation of the route, and input fields for bus stop and bus selection.
- Backend rendering static information for displaying stops and bus options to users; also executing the machine learning models and returning predictions related to journey duration.
- Database storing Dublin Bus and Dublin weather data; accessing relevant data for frontend operations via backend.
- Data analytics to analyse historic Dublin Bus and Dublin weather data; train Machine Learning models for generating a prediction of journey duration.

3.1.2 Team Analysis

As the team consisted of 4 members, we decided to allocate 3 resources to development and 1 resource for project documentation and team operations. All the team members discussed their area of expertise, choice of organizational role, and amount of effort they are willing to put in each component of the project [6]. This proved effective while planning the project life cycle.

3.2 Strategy for Team Management

3.2.1 AGILE Methodology

All members were experienced with AGILE methodology; hence the team decided to proceed with AGILE development. AGILE development works on the motto of continuous development of deliverable software and regular face to face meetings. As remote collaboration and development were new to some team members, the team agreed to have agenda meetings twice a week; for brainstorming on design and analytics challenges.

3.2.2 Project Management

Communication

- Trello is an interactive kanban board developed by Atlassian. SCRUM master organized project sprints into Trello lists. Lists consisted of Trello cards for individual stories; which were assigned to team members. Trello features like tagging, labels for project components, checklists, attachment facility, and due dates ensured that no issue went untracked.
- Google Meet provided quality video conferring facilities for SCRUM and agenda meetings, as well as meetings with project mentors. It is well integrated with Google Calendar, and easy to use; with no downtime.

Data Sharing Platform

- Google Drive was used for creating and maintaining project documentation. Shared editing avoided documentation versioning and inconsistencies. Maintaining team attendance sheet, testing logs, presentations on google drive was easier due to G Suite integration.

Version Control

- GitHub is a version control tool for software development; heavily used in the software industry. The team maintained separate frontend and backend git repositories on GitHub adhering to industrial practice [7]. Collaborative development and updating deployed applications were efficiently done with GitHub.

3.2.3 Sprint Architecture

The whole project was logically split in 4 major sprints.

- Planning and Database Setup: Sprint 1 focussed on understanding the dataset. For technical understanding; the team created proof of concepts for frontend and backend stacks; React.js-Redux.js, and Django respectively.
- Django and Web UI Design: Sprint 2 was designed to implement and test the frontend-backend integration. This sprint also allowed resources who worked on data analytics to consult mentors continuously for finalizing the input features and choosing machine learning algorithms.
- Data Analytics: Sprint 3 was designed to test the integration of trained prediction models with the deployed web application.
- User experience and Improvements: Sprint 4 focussed on resolving bugs, improvement of a web application, robustness, and addition of innovative features like Weather Prediction, Fare Estimator, Leap Card Integration, and Twitter Feed.

Chapter 4: **Technical Approach**

4.1 Architecture

This web application consisted of two main components i.e. frontend and backend. The frontend and backend were two separate standalone applications with different technologies. So, these two applications could interact with one another and other third-party providers using RESTful Web Services. Whenever the frontend requested a service, the backend communicated with the database or machine learning models to provide an appropriate response. Moreover, a CRON job was executed by the backend which interacted with the OpenWeather API every 60 minutes for scraping weather data predictions. This open source API offered weather predictions for 5 days [8].

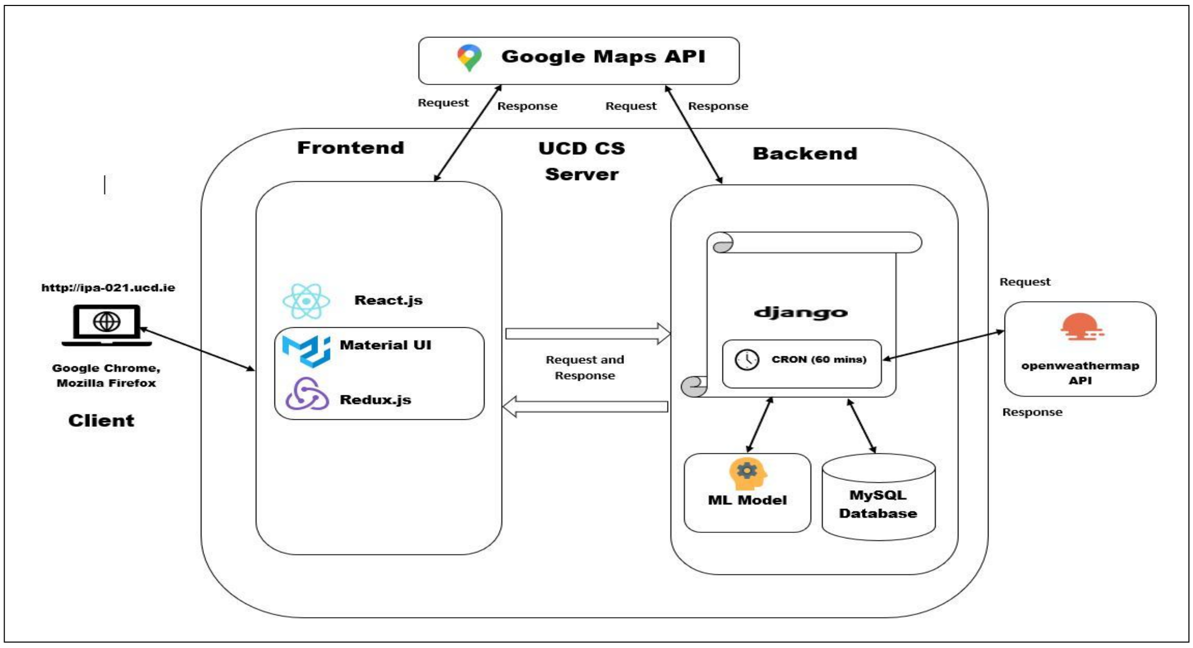


Figure 4.1: System Architecture

4.2 Technical Stack

4.2.1 Frontend

Two of the most popular technologies found were Angular and React.js. React.js was chosen given its ease of learning and faster page loading capabilities. Moreover, manifesting the present website stack for a mobile application would be easier with React native; which is based on React [9]. Along with, Redux.js is used for maintaining JavaScript state which increases the efficiency of the website. For the user interface, the Material UI framework was chosen due to its usage of Google's Material Design language and ease of implementation in React.js. Learning a new JavaScript technology stack was helpful to us rather than using "drag and drop" web design platforms like WordPress, GoDaddy, Wix, etc.

4.2.2 Backend

Python is a powerful server-side programming language offering seamless machine learning integration along with web app development. Django was the ideal Python framework due to its "Batteries Included" approach. We utilized developer-oriented capabilities of Django such as routing of URLs, creation of database migration schema, built-in ORM system, etc. For serving all the client-side requests, the Django REST framework was used; with its powerful and flexible toolkit, serialization, authentication policies, etc. Lastly, for saving the user's data, MySQL was finalized due to its familiarity and capability to handle large amounts of data. We considered using "AI as a service" offered by cloud platforms like AWS, Azure [10][11]. But considering data protection policy for the Dublin Bus 2018 data; we preferred developing prediction models locally.

4.2.3 Data Analytics

The team performed the Extract - Transform - Load (ETL) cycle on hourly weather data of Dublin city for the entire 2018 from OpenWeatherMap API and Dublin Bus 2018 database provided for this project only. For the ETL cycle, the team used python scripts on server and industry-standard Pandas data frames with Jupyter notebooks on local machines. The main database file was of size 11 GB. It was divided into 11 subfiles of 1 GB each using Bash command line interface; which facilitated data handling under physical memory constraints. Exploratory data analytics involved Tableau visualizations and gathering sources of truths for each feature through anomaly detection. This helped in creating a data frame for training prediction models. The team experimented with various prediction algorithms like Light Gradient Boost Machine and feature encoders like nominal encoders. We opted for Linear Regression prediction models considering its lightweight nature and metric performance. To make the prediction model transportable, pickle library was used. This enabled resources to work on data analytics to ship lightweight models to the production server. For the 2018 Dublin traffic data, the team attempted to speak with sources like TomTom [12][13]; but efforts did not conclude.

Chapter 5: **Testing and Evaluation**

5.1 Testing Strategies

5.1.1 Unit Testing

Unit testing was performed by developers to test individual components of the application. All known functionalities in a component were checked for true positive as well as true negative results. Error messages and deterministic data storage were also validated. To ensure the validity of requests and responses exchanged between frontend and backend; API testing was performed using Postman. For example, while developing the Registration component; all field validations were checked along with application environmental conditions like backend failure, internet failure, etc. Unit testing captured the issues in the early stages and helped in reducing the bugs.

5.1.2 Peer Testing

The team was homogeneous and all resources had to work on the functionalities of all components. Hence, we peer-reviewed features developed by each other. A bug sheet was maintained to track bugs from identification until resolution [14]. Additionally, we followed the Manual White Box Testing approach [15]; focusing on the integrity of logic. This scrutiny helped in improving application efficiency by making decisions as follows:

- Indexing: For efficient database querying, we decided to employ Indexing while defining backend models.
- Code Reuse: Method for visualization of a route on the map was reused by calling it from different frontend sections.
- Redux store: To prevent redundant requests to the backend for re-fetching static data; a store saved static data and the frontend used it as required, which increased the speed of the website.

5.2 Results and Evaluation

Acceptance Testing was performed on the deployed web application by all team members. We analyzed lighthouse reports [16] and fixed critical bugs like unused frontend imports, handling console warnings. To analyze performance of prediction models, metrics for models were logged in validation report [17]. We observed indicators like R2 for overfitting, Mean Absolute Error for accuracy. We observed that "X" buses had sparse timetables and error measures for these buses also had large deviations. Whereas, most of the regular buses having normal frequency had error bounds within 5 minutes. More efforts and experimentation with hyperparameter tuning can lead to better prediction performance.

Bibliography

[1] Dublin Bus Annual Report [online] Available at:
https://www.dublinbus.ie/Global/Bus-Atha-Cliath-2018-Annual-Report.pdf [Accessed 18 August 2020].

[2] dublinbus.ie. 2020. *RTPI - Dublin Bus.* [online] Available at:
https://www.dublinbus.ie/RTPI/ [Accessed 17 August 2020].

[3] Fabrikant, A., 2020. Predicting Bus Delays With Machine Learning. [online] Google AI Blog. Available at:
https://ai.googleblog.com/2019/06/predicting-bus-delays-with-machine.html [Accessed 17 August 2020].

[4] leapcard.ie. 2020. [online] Available at:
https://www.leapcard.ie/Home/index.html [Accessed 17 August 2020].

[5] dublinbus.ie. 2020. Adult - Dublin Bus. [online] Available at:
https://www.dublinbus.ie/Fares-and-Tickets/Adult/ [Accessed 17 August 2020].

[6] Google Docs. 2020. Team Intro.Xlsx. [online] Available at:
https://drive.google.com/file/d/11XtCnegHeGJ15HcfsbuvDYfcsEc8yzqL/view?usp=sharing [Accessed 17 August 2020].

[7] Mason, J., 2020. The Front-End Deserves Its Own Repository. [online] Medium. Available at:
https://medium.com/@joeyvmason/the-front-end-deserves-its-own-repository-86fe382b7d37 [Accessed 17 August 2020].

[8] Openweathermap.org. 2020. 5 Day Weather Forecast - Openweathermap. [online] Available at:
https://openweathermap.org/forecast5 [Accessed 17 August 2020].

[9] Faraday, G., 2020. Converting A React App To React Native. [online] Medium. Available at:
https://medium.com/@gwen-faraday/converting-a-react-app-to-react-native-d7df17968fc6 [Accessed 17 August 2020].

[10] Amazon Web Services, Inc. 2020. Time Series Forecasting — Machine Learning — Amazon Forecast. [online] Available at:
https://aws.amazon.com/forecast/?c=mlsec=srv [Accessed 17 August 2020].

[11] Azure.microsoft.com. 2020. Azure Machine Learning — Microsoft Azure. [online] Available at:
https://azure.microsoft.com/en-us/services/machine-learning/ [Accessed 17 August 2020].

[12] Google Docs. 2020. Tomtom Query.Pdf. [online] Available at:
https://drive.google.com/file/d/1RCkCoXAytza6fZPMFhpY7nukBS8bM4gY/view?usp=sharing [Accessed 17 August 2020].

[13] Google Docs. 2020. Tomtom Response.Pdf. [online] Available at:
https://drive.google.com/file/d/1LC-qScjZA5kWLhD5RXP9YBt4b74Wngh3/view?usp=sharing [Accessed 17 August 2020].

[14] Google Docs. 2020. Testing.Xlsx. [online] Available at:
https://drive.google.com/file/d/1kpebE5V1O4a3N6rQisO2lfYHTzk0VMv5/view?usp=sharing [Accessed 17 August 2020].

[15] Softwaretestinghelp.com. 2020. White Box Testing: A Complete Guide With Techniques, Examples, Tools. [online] Available at:
https://www.softwaretestinghelp.com/white-box-testing-techniques-with-example/ [Accessed 17 August 2020].

[16] Lighthousereport9aug.Pdf. [online] Available at:
https://drive.google.com/file/d/1YVY02vZDVIzwM0N9tWuuTeys4mdbbBaw/view?usp=sharing [Accessed 18 August 2020].

[17] Validationresults.Csv. [online] Available at:
https://drive.google.com/file/d/1DUo2fk3srCm8c3cKuwmNu37cyepSshER/view?usp=sharing [Accessed 18 August 2020].