Addis Ababa Institute of Technology

School of Information Technology and Engineering

Department of Software Engineering

Location Intelligence: Supermarket Site Selection

All In One Documentation

Team Members

Eyosias Samson	ATR/0484/09
Gemmechu Mohammed	ATR/1432/09
Khalid Sultan	ATR/8444/09
Tsedeniya Solomon	ATR/9796/09

Advisor: Mr. Nebiat Fikru

Date: September/2021

ACKNOWLEDGMENT

Many people have contributed to the development of this concept into what it is today. As a result, we'd like to express our gratitude to these individuals for their assistance. First and foremost, we would like to express our heartfelt gratitude to our advisor, Nebiat Fikru, for his unwavering support of our final year project, as well as his motivation and vast knowledge.

We would like to thank the School of Information Technology and Engineering, as well as the project committee and for allowing us to design and implement this project.

Dr Degife Tibebe, our former Geographic Information System teacher, deserves our sincere appreciation for assisting us in gathering data. And we would like to thank Mr Amanuel Negash, our teacher for his guidance throughout the project.

Finally, but not least, we would like to thank our family for their enormous support in our lives and for encouraging us.

Table of Contents

List of Figures

List of Tables

ACRONYMS

COVID-19 - CoronaVirus disease of 2019

GIS - Geographic Information System

PhD - Doctor of Philosophy

API - Application Programming Interface

CRISP-DM - Cross-industry standard process for data mining

ABSTRACT

If you are launching a new business or expanding an existing one, the location is a critical factor in determining whether the venture will prosper or fail[1]. People will spend a significant amount of time and resources researching where they can open their company, and they will frequently have a narrow selection with little insight on how their business will do. This project will try to address this issue by the use of location intelligence, which involves gathering and processing a wide range of geospatial and demographic data and transforming it into strategic insights to solve the problem. This project focuses on supermarkets and will help the business owners in making an informed decision by presenting choices and explaining why it will be the right location for their business.

1. Chapter 1: Introduction

1.1 Background

Ethiopia is one of the fastest developing countries despite the recent COVID-19 outbreak with a **6.1%** growth rate [2]. As new business institutions open up throughout the city, businesses are capitalizing on this growth. This provides motivation and incentive for every-entrepreneur to take an initiative and make their dreams a success. But in reality, most of them don't survive because of different reasons, one of them being location, location comes up as the sole disruptor simply because the location chosen wasn't ideal for the type of service they were providing.

Throughout the world, businesses fail at different rates but most of the underlying reasons aren't unique. In the United States of America, for example, 34% most flexible businesses expand or start anew, Retail Stores are a prime example [3].

In Ethiopia using location intelligence to choose a viable location for setting up your business is fairly uncommon. The common practice is physically going to visit locations and scouting if they are viable or not. Maps are rarely utilized. In Addition to that, most people believe maps as static data where the only information to extract from them is the location for places. So they rather rely on their own experience and knowledge. This project will mainly focus on how to improve Supermarkets Site selection by using maps and applying location intelligence. Similar products have been done globally which are explained in section 1.2. Some of them use pure GIS to solve this issue, while others focus on specific company examples.

1.2 The Existing System

There are a couple of research papers and products that are trying to solve these problems. All these solutions are outside of Ethiopia.

- **Targomo [4]**

  Targomo was founded in 2013 and is based in Berlin and Potsdam. The platform is being used in: real estate and retail companies to forecast the potential of locations and optimize location networks, offers or delivery routes. Public transport companies to optimize their routes and transit network or promote green mobility.

- **Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API** [5]

  A research paper was done by PhD candidates from the Department of Econometrics, Istanbul, Turkey. The research uses Google Places API services to get the data for nearby places. The system estimates the relation between environmental properties and existing stores and their financial rankings using Machine learning.

- **Azure Maps** [6]

  Azure Maps pulls in data on locations, competitors, traffic, and public transit through Microsoft partnerships with TomTom and Moovit to analyze catchment. Azure Maps also ingests data from any data provider including companies that offer human mobility data. Catchment analysis can help a retailer analyze business disruptors, decide where best to open a store, or identify opportunities for expanding business.

1.3 Statement of the Problem

Element	Description
The problem of Manual location searching	Businesses in search of a new location will have to do manual location searching which consumes Time, energy and money. realistically this method can't search for every location as a result the business might miss a good business location.
Affects	Any business that depends on physical location. As an example, consider the supermarket.
And results of Location on Businesses	The impact of physical location on business is very high. It could mean having a blossoming business or going out of business.
Benefits of a solution using Location Intelligence	The proposed solution is a system that will analyze a location concerning its environment and rank the location according to its profitability. - Saves time, money and energy - gives more option to choose from

1.4 Objective of the Project

1.4.1 General Objective

The goal of this project is to develop an intelligent web-based system for supermarket new site selection.

1.4.2 Specific Objective

- Data gathering and preparation
- Algorithm selection for the machine learning component
- Machine learning algorithm evaluation and training
- User interface design and implementation

1.5 Proposed System

In this project, we propose to develop an intelligent web-based system for supermarket new site selection. We will create web applications that will allow users to interact with the system and view the results. The system allows users to input the preferred locations through map interface and receive ratings for each location. They can also view location and demographic details about the location.

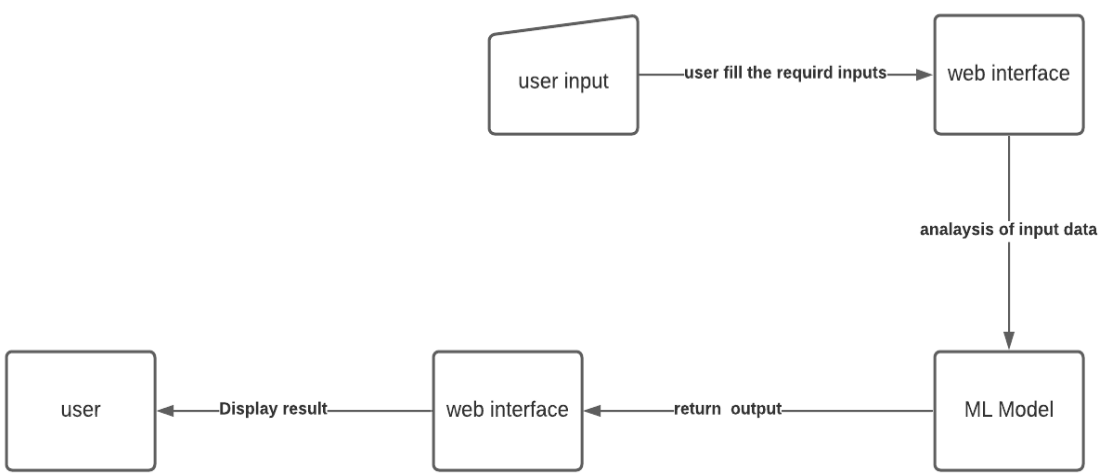

Figure 1 Data flow Diagram

1.6 Feasibility Study

1.6.1 Economic Feasibility

1.6.1.1 *Developmental cost*

Table 1 Developmental cost

Reason	Cost (For Five Months Total)
Transportation	600 ETB
Internet and Phone Call Fees	5000 ETB
Google Maps API	Free (up to 100,00 requests)
Total	5600 ETB

Other than the costs described above, the system will not require additional cost to develop as it is not reliant on any external hardware requirement or paid software services.

1.6.1.2 *Operational Cost*

We'll be hosting and maintaining the system on a free platform and it doesn't need manpower to operate it once it's developed. So we can say there will be no operational cost that can be predicted for the project at this point.

1.6.2 Technical Feasibility

Our team is composed of members who are adept at Machine Learning and Full Stack Web Development. We'll be using free and open-source development as described in 1.8.3. But there are a few challenges facing the development of this project.

- We may not get demographic data from sub-city administrative regions as well as foot traffic data around the city.
- Returning a specific location data will be difficult as the granularity at that level causes the complexity and difficulty to exceed well beyond the scope of the project. Instead, it will return area-level data based on a certain radius chosen.

1.6.3 Schedule Feasibility

Despite the difficulty that is associated with this project and that the duration for projects this year have been cut down significantly, a proper schedule and continuous progress should provide enough time for the project to reach a demonstrable stage. We will use the time management policy listed in the time management plan section to finish the project on time.

1.7 Scope

The following features are the most important and define the entire product out of all the features that this product is supposed to have.

- The entire system will be focusing on Supermarkets.
- It will also be restricted to the city of Addis Ababa only.
- It will be web-based and it will include a map interface to display the appropriate locations based on the user's input.
- The interface will also display the reasoning behind why it chose those locations. It will show the results of location intelligence based on several factors like nearby competitors, places of interests and others that are weighted and given values that describe how preferable a certain location is.

1.8 Methodology

1.8.1 Data source and collection

Most of the data for this project will be collected from the Google Maps Platform. Additional data not found from Google Maps API that is a necessary criterion for the project like demographic data will be taken from Dr Degefie's personal work dataset given to us.

1.8.2 Software Development Lifecycle

The plan for the development of the system takes into account the fact this is a Machine learning project with a defined set of core functionalities as a result the project will follow CRISP-DM methodology.

1.8.3 Development Tools

The tools that are necessary for the project are the following.

- Google Maps - to extract data that deals with the criteria the retail industry values
- Javascript Framework - to build the user interface
- Python - for data analysis and machine learning

1.8.4 Testing

Internal testing is done throughout each iteration of the project development phases. while external testing is done by multiple iterations of understanding the business problem by asking questions, data acquisition from multiple sources, data cleaning, feature engineering and modelling.

1.9 Project Management plan

1.9.1 Time Management Plan

	ACTIVITIES	START	WD	DUE	%
	Project Conception and Initiation:	23/Mar	18d	15/Apr	100%
2	Brainstorming ideas	23/Mar	14d	09/Apr	100%
3	Requirement Gathering	10/Apr	4d	15/Apr	100%
4	Title Approval	10/Apr	4d	15/Apr	100%
	Project Definition and Planning:	14/Apr	22d	13/May	39%
6	Scope and Goal Setting	14/Apr	3d	16/Apr	100%
7	Communication Plan	15/Apr	2d	16/Apr	100%
8	Proposal Development/Defen...	15/Apr	6d	22/Apr	80%
9	Requirement Analysis	22/Apr	6d	29/Apr	0%
10	SRS Development	04/May	8d	13/May	0%
	Project Conception and Initiation:	06/May	51d	15/Jul	0%
12	Data Collection & Preparation	06/May	6d	13/May	0%
13	Machine Learning Algorithm ...	13/May	4d	18/May	0%
14	Evaluation & Training of Model	18/May	24d	18/Jun	0%
15	User Interface Development	14/Jun	19d	08/Jul	0%
16	Integration	08/Jul	6d	15/Jul	0%
	Project Performance / Monitoring:	16/Jul	53d	28/Sep	0%
18	Parameter Tuning	16/Jul	12d	02/Aug	0%
19	Integration Testing	02/Aug	17d	24/Aug	0%
20	Project Finalization	13/Aug	33d	28/Sep	0%

Figure 2 Time Management Plan

1.9.2 Quality Management Plan

The most significant risk for this project is obtaining accurate data on the required features. Because we will be using geospatial data, obtaining the correct data may be difficult. To solve this problem, we will use Google Place API to extract data using the free version.

The other is that using population data as a feature may present some difficulties because it requires labelling and extracting the necessary data from statistical data. In this regard, we will collaborate with Dr Degife, who taught us GIS to get accurate and well-structured data.

To explain the machine learning model to users. We will create a web interface to allow users to interact with the system. The user interface will be simple to use even for inexperienced users.

Unit testing will be used to test the system at the unit level. After we finish building the system, we will conduct an integration test to ensure that all of the system's modules function as expected.

1.9.3 Communication Management Plan

We will communicate about the project using the following tools:

- **Google Meet:** We will use this tool to hold weekly meetings with the members of the group and the advisor.
- **Telegram**: Alternative method of communication with group members for short idea sharing
- **Google Drive**: we will use to share project documents
- **Trello**: We will use it to break down tasks and keep track of progress.
- **GitHub**: To share project codes.
- **Phone calls**: In the event that we are unable to access other tools, we will use this tool to communicate with the group members.

Table 2 Communication Management Plan

Type of Communication	Method / Tool	Frequency/ Schedule	Information	Participants / Responsibilities
Internal Communication:				
Project Meetings	Direct Meetings, Google Meet, Telegram, Phone call, Trello	Weekly and on event	Project status, problems, risks, changed requirements	Project Manager Project Team
Sharing of project data	Google Drive, GitHub	When available	All project documentation and reports	Project Manager Project Team Members
Milestone Meetings	Direct Meetings, Google Meet	Before milestones	Project status (progress)	Project Manager Sub-project Manager

Final Project Meeting	Direct Meetings, Google Meet	M6	Wrap-up Experiences	Project Manager Project Team
External Communication and Reporting:				
Project Report	Google Drive, Microsoft Word	Monthly	Project status - progress - forecast - risks	Project Manager Sub-Project Managers

2. Chapter 2: Requirement Analysis

2.1 Introduction

In this chapter we outline product perspective, product functions, user characteristics, general constraints and assumptions and dependencies of the system to be developed. It also includes external interface requirements, use cases, non-functional requirement, inverse requirement, user interface, design constraints and logical database requirement and other requirements of the system.

2.2 Product Perspective

Currently, supermarket site selection is done manually, with people going to all of the possible locations to make a decision. Using a manual site selection has its own drawbacks. Among them are the following:

- When going around looking for a good location, a lot of labor is required because people must go from place to place looking for the best site, which can be very exhausting.
- When manually selecting sites, it can be costly to hire people to do the searching, and going from place to place incurs travel costs.
- When looking for the best supermarket location, you must visit a number of locations before deciding on the best one, which wastes time.

There are applications and studies that use location intelligence to predict business locations. Those applications, however, are not available in Ethiopia. The main reason is that the majority of the applications we reviewed on data that is inaccessible here in Ethiopia like foot traffic data or company profit. Another reason we can also consider is that most of these applications are not built with Africa or more specifically Ethiopia and other countries where data is scarce.

Some of the applications reviewed include the following:

I. **Targomo**: Real Estate Search [4]

Targomo's Real Estate Search is a location intelligence platform used by both real estate companies to forecast potential locations, and home seekers to search for houses tailored and relevant to their needs.

Real Estate Search analyzes a wide range of locations that includes offices, gyms,universities, and hospitals and different ways to get there (car, bike,public transport,walkway) of a given country and recommends real estate companies potential spots for real estate development by the use of machine learning.

In return, home seekers include travel times, commute routes, neighborhood preferences, as their search option and Targomo is able to make recommendations of potential housing that fit the attributes selected by the user.

Targomo's web services that are available in all continents except Africa.

II. **Retail Location Analytics with Azure Maps** [6]

Azure Maps can be used by Retailers to better understand a location and analyze business drivers and disruptors from a spatial perspective by performing Catchment analysis[6] and making use of the Azure's Route Range API.

A Catchment analysis is part of identifying whether it is viable to open a new retail store or outlet in a particular location. The analysis considers the following factors.

a. Socio-demographics - analysing consumers by age, life stage,education and income to help identify what customers look like, what they might be interested in, their disposable income and buying power.
b. Infrastructure - the surrounding transport network of roads, walkways and public transport indicate how accessible a location is. This accessibility will influence how far away your target audience might live and can impact how far they are prepared to travel
c. Residential & workforce population - looks at both residential and workforce population. The number of businesses, size and industry type will be important on the sales volume of a retail store.
d. Competitor analysis - looks at the local businesses within an area , the quantity, products and services offered and price point. These factors identify the potential level of demand of products and services and can highlight a retail service's pricing or category ranging strategy.

One of the biggest drawbacks of this application and is agreed upon even by the developers is that users see Azure maps as a service to just render maps and places. The Retail Location Analytics service is presented as a feature more than an independant marketable service and thus seems hidden from the common user .

III. **Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API** [5]

A research paper published in December 2013 by PhD candidates from the Department of Econometrics, Istanbul, Turkey, describes a system that estimates the relation between environmental properties and existing stores and their financial rankings using Machine learning.

The research uses Google Places API services to get the data for nearby places. The system describes the geospatial attributes that affect a retail success as environmental properties. These environmental properties include bus stations, restaurants, banks, hospitals, religious institutions, entertainment venues, public transport stations, and schools.

The proposed method is based on construction of an Artificial Neural Network with a hidden layer of 120 Neurons that takes in environment properties as an input and returns rankings as output. This network has been trained multiple times, with the final network reported in 0.74 seconds with a network error of 0.00009. This small network error shows that the network fits the training data well but forecasting the data of the network should be investigated. For this purpose, looking at the Mean Absolute Deviations,Mean Square Methods, and Mean Absolute Deviations are quite small compared to the maximum possible values in case of worst forecasting performance.

The research describes that after reviews by experts, the method produces reasonable estimates of Ranking of candidate stores. The method also reveals similarities between the candidates and currently available stores.

The aim of this project is to create a system that will provide supermarket site selection using patterns recognized from nearby locations of currently existing supermarkets in Addis Ababa.

Unlike currently existing systems, this project has a narrower scope and targets only Addis Ababa supermarkets.

2.3 Product Functions

The system is able to rate and rank user suggested sites for a supermarket based on nearby places and demographic data. This system will provide ways for viewing top rated sites and provide details on why each site received a specific rate.

The count of nearby places that we consider as parameters are:

● Restaurants	● Schools	● Gas Stations
● Supermarkets	● Banks	● Bus Station
● Hospital	● Religious Institutions(Mosques/ Churches)	● ATM
● Pharmacy	● Train Station	

2.4 User Characteristics

There is one type of user which uses the system. The user is expected to have basic computer skills and can access the internet.

2.5 General Constraints

- The up-to-date google maps - most of the data needed for the project comes from Google Maps Places API. The project accuracy dominantly depends on how accurate google map's data is.

2.6 Assumptions and Dependencies

The user is familiar with the fundamentals of computer operation and can access the internet via a browser.

2.7 User Interfaces

Table 3 UI-01 Home Page

ID	**UI-01**
Name	Home page

Description	User interface which displays a map interface to users.

Figure 3 UI-01 - Home Page

Table 4 : UI-02 Preferred Locations

ID	**UI-02**
Name	Preferred Locations
Description	Displays the user's chosen preferred location.

Figure 4 UI-02 - Preferred Locations

Table 5 UI-03 Analysis Result

ID	**UI-03**
Name	Analysis Result
Description	Displays the rated locations in ascending order.

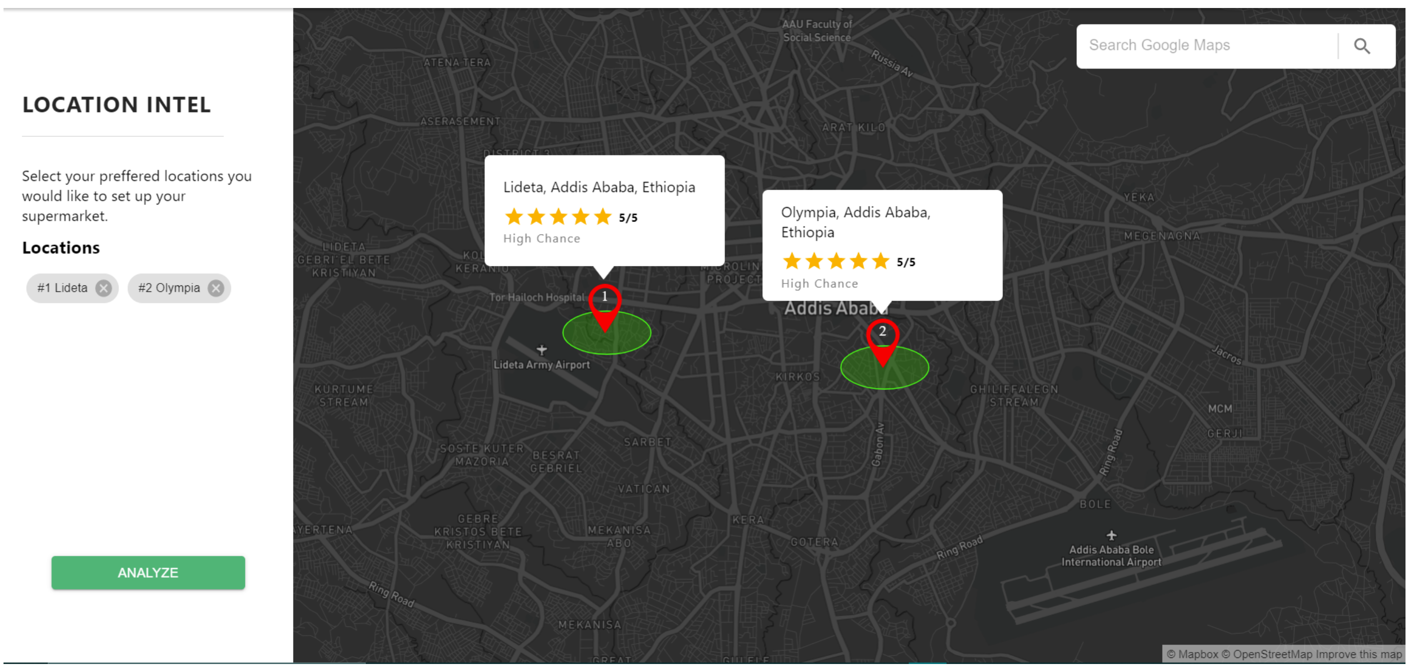

Figure 5 UI-03 - Analysis Result

Table 6 UI-04 - Location Details

ID	**UI-04**
Name	Location Details
Description	Displays the details of the analyzed locations.

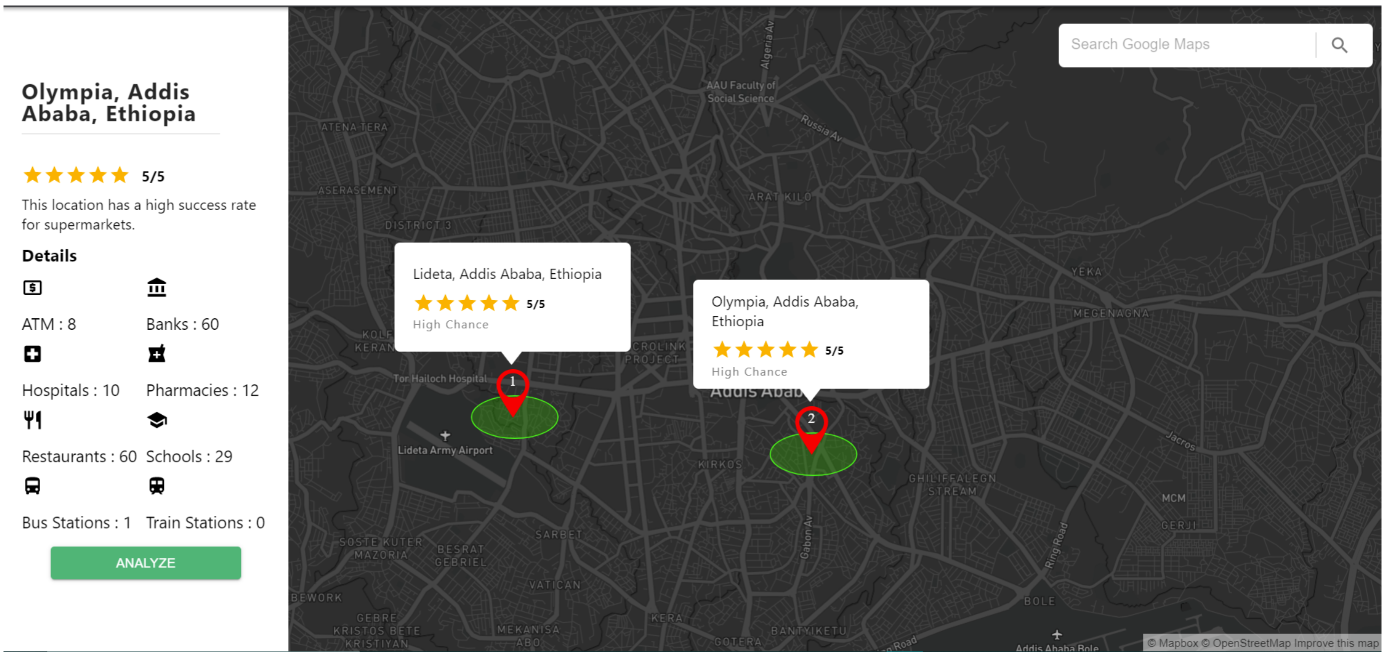

Figure 6 UI-04 - Location Details

2.8 Hardware Interfaces

The system has no hardware interface requirements.

2.9 Software Interfaces

The system will use Javascript Frameworks to build the user interface. We also will use Python frameworks to build the machine learning model. The system uses both Google maps and Mapbox to display and utilize map details. We'll also use MongoDB to serve as the database for caching the predicted results.

2.10 Communications Interfaces

- **Web Interface**

  The network would be used to access the application. The web site will provide complete access to all features.

- **HTTP protocol**

  The HTTP protocol will be used to communicate across the network by the System. Furthermore, this facilitates communication between clients and servers.

2.11 Functional Requirements

2.11.1 FR-1 Select Preferred Locations

Table 7 FR-01 - Select Preferred Locations

ID	**FR-1**
Introduction	The system shall allow the user to select preferred locations by navigating on a map
Inputs	None
Preprocessing	None
Outputs	Location coordinates
Error handling	None

2.11.2 FR-2 Analyze Locations

Table 8 FR-02 - Analyze Locations

ID	**FR-2**
Introduction	The system shall analyze selected coordinates by gathering information from nearby places and demographic data.
Inputs	None
Preprocessing	FR-1

Outputs	A collection of preprocessed information for selected coordinates
Error handling	None

2.11.3 FR-3 Rate Locations

Table 9 FR-03 - Rate Locations

ID	**FR-3**
Introduction	The system shall predict the rating of a location using the ML model.
Inputs	location demographic and nearby place data
Preprocessing	FR-2
Outputs	A set of ratings for selected locations
Error handling	None

2.11.4 FR-4 Location Details

Table 10 FR-04 - Location Details

ID	**FR-4**
Introduction	The system shall allow the user to see a location's detail.
Inputs	None
Preprocessing	FR-3
Outputs	Detail of a location
Error handling	None

2.12 Use Cases

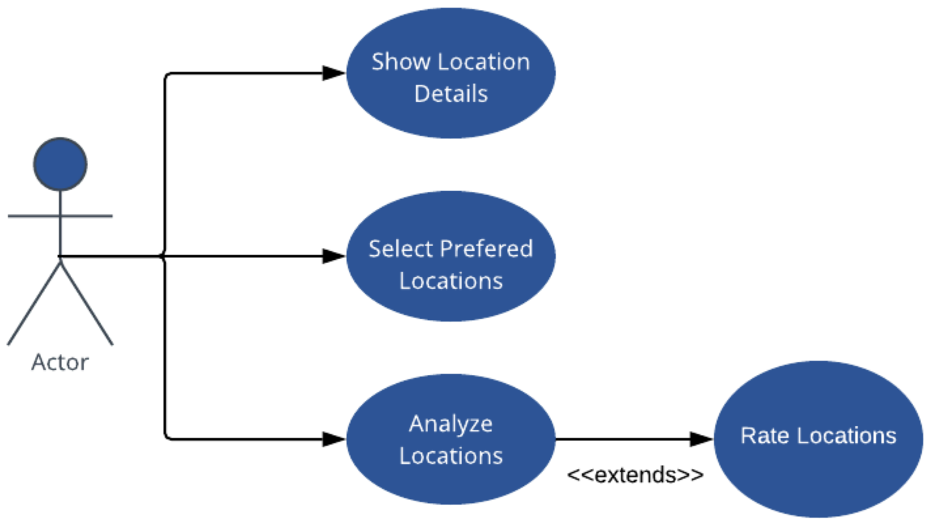

Figure 7 Use Case Diagram

2.12.1 Use Case 01

Table 11 UC-01 - Select Preferred Locations

ID	UC-01
Name	Select Preferred Locations
Description	Allow users to select their preferred location by hovering over the Map
Actors	User
Pre Condition	None
Post Condition	Display selected locations
Triggers	The user wants to input its ideal supermarket locations
Course of Action	1. User opens up website 2. System displays Home page user interface 3. User pin a point on the map 4. The system displays the pinned points in the Preferred Locations user interface (UI-02) 5. Use case ended

2.12.2 Use Case 02

Table 12 UC-02 - Analyze Locations

ID	UC-02
Name	Analyze Locations
Description	Allows users to rate the selected locations
Actors	User
Pre Condition	Selecting Preferred Locations (Use case - 01)
Post Condition	Display rates of locations
Triggers	The user wants to know the rates of selected locations
Course of Action	1. System displays Preferred Locations user interface 2. User clicks the analyze button 3. The system start analyzing the locations 4. The system displays the rates of locations in the Analysis Result user interface(UI-03) 5. Use case ended
Alternative course of Action	2.a. User provides invalid location 2.a.1. The system display invalid location 2.a.2. The system starts back at Home page user interface

2.12.3 Use Case 03

Table 13 UC-03 - Rating Locations

ID	UC-03
Name	Rate locations
Description	The system predict the rating of a location
Actors	System
Pre Condition	Analyze Locations (Use case - 02)
Post Condition	Determine the rating of a location
Triggers	The user clicks analyze button
Course of Action	1. The system requests the backend if the location exists in the cache 2. If not, The system searches for nearby places

	3. The system retrieves demographic data 4. The Model predicts the rate 5. The system displays the rates of the locations in Analysis Result user interface(UI-03) 6. Use case ended
Alternative Course of Action	2a. Location is found in the cache 2a.1. The system returns the cached predicted results 2a.2. The system displays the rates of the locations in Analysis Result user interface(UI-03) 2a.3. Use case ended

2.12.4 Use Case 04

Table 14 UC-04 - Location Details

ID	UC-04
Name	Show Location details
Description	Users can see details of the rated location
Actors	User
Pre Condition	Analyzing Locations (use case 02)
Post Condition	Display details of rated location
Triggers	The user wants to find details of the rated location
Course of Action	1. System displays Analysis Result user interface 2. User clicks location detail button 3. The system display the details of the location in Location Details user interface 4. Use case ended

2.13 Non-Functional Requirements

2.13.1 Performance

- 90% of the transactions shall be processed within 5 seconds.

2.13.2 Reliability

- The system should be able support at least 2000 requests without failing on its server. It should have to display the precise locations and display them appropriately on a map.

2.13.3 Availability

- The system will be available 24 hours a day, 7 days a week.

2.13.4 Security

- The system uses HTTPS protocol through SSL/TLS Certificates to avoid HTTP attacks.

2.13.5 Maintainability

- The system attains its maintainability by using modular programming and little dependencies between the modules. It uses Github for version tracking and deployment.

2.13.6 Portability

- Because the system is designed for web applications, it should be accessible from any device that supports HTML, CSS, and JavaScript. To support a wide range of browsers, the system will be designed in a responsive manner.

2.14 Inverse Requirements

- The application does not take into account or aim to improve a supermarket's internal marketing strategies such as customer retention schemes and giveaways.

2.15 Design Constraints

- Google Maps Places' API can only retrieve a maximum of 60 nearby places
- If the system utilizes Google Maps - Architecture, the model will automatically update.

2.16 Logical Database Requirements

Table 15 Logical Database Requirements

Table	**Description**	**Attributes**
Cache	Holds the results of previously predicted data for a certain time as a temporary cache to optimize performance	latitude:float longitude:float rating:float details:list<<String>>

2.17 Other Requirements

- **Training-related Requirements**

  This system can be used by anyone with a basic understanding of computers. As a result, there is no training for the system. Furthermore, we can provide end users with documentation that explains how the system works.

- **Packaging Requirements**

  The project is a web-based program that includes a variety of file formats that are associated with the web discipline and Machine learning model. To name a couple, the ".JS" and "Read me" text files that will be packed.

- **Legal Requirements**

  The system will be available on GITHUB under an MIT license. We chose this license over other open-source licenses because it is a short and easy permissive license with only the protection of copyright and license notices as conditions. Licensed works, revisions, and larger works can be distributed with or without source code under different terms.

2.18 Change Management Process

Requirements may change during the development course of the project. These changes are applied only after going through the following change management process.

A team member proposes a requirement change. The proposed change should include scope, reason and impact for the change.

- The pros and cons of the change will be examined by the team.
- The requirement will be submitted to the advisor for approval.
- The advisor will decide if the change is relevant
- The requirement change will be added to this document.

3. Chapter 3: System Design

3.1 General Overview

The purpose of this Supermarket Selection Application is to enable business owners and entrepreneurs to facilitate site selection using demographic data and data collected around supermarkets in Addis Ababa. It's limited within the scope of focusing specifically on the city of Addis Ababa and the field of Supermarket locations. It will give the users the ability to choose their preferred locations.

The system will have four main components: the Web application, the ML model, Backend and the Google Maps Places API. The Web application will have the main canvas that users can provide information about the preferred locations. It also gives the functionality for more information on predicted insights. The backend communicates with the model and database to facilitate the caching process. The ML model does most of the work; it processes the given locations and rates them based on previously learned data.

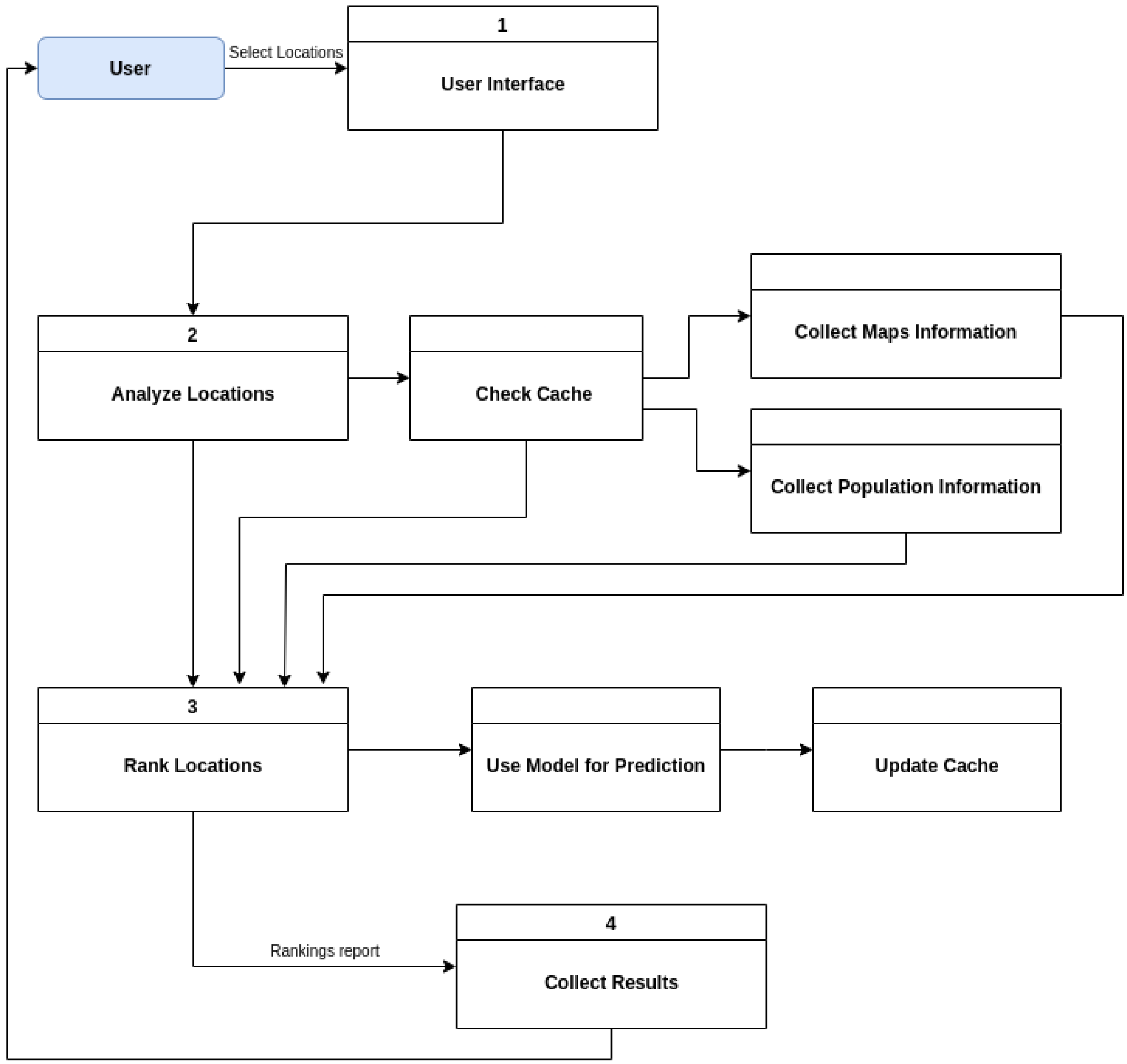

Figure 8 System Overview

3.2 Development Methods & Contingencies

This project will follow the CRISP-DM methodology. This framework is best fit for carrying out data mining projects and sets out activities to be performed into six phases. The successful completion of a phase initiates the execution of the subsequent activity. CRISP-DM includes iterations of revisiting previous steps until success or completion criteria are met.

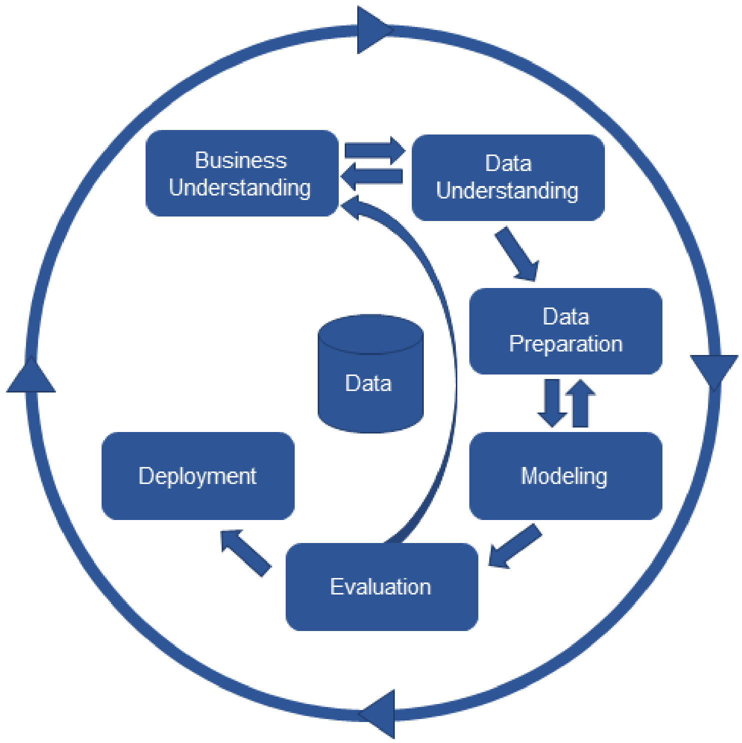

Figure 9 CRISP-DM Cycle

This project will be done in the phases of the following.

- Business understanding - Our first phase of the project includes finding out what geospatial attributes make a successful supermarket. We will list and research until a set of attributes is agreed upon.
- Data understanding - The second phase consists of understanding our sources and meaning of our data.
- Data Preparation - The next phase consists of cleaning, labeling and feature engineering the geospatial data we have in order to filter and select the attributes we agreed upon.
- Modeling - Fourth phase consists of training the model with the ANN (Artificial Neural Network) algorithm and K-means clustering.
- Evaluation - Assess the degree at which the model meets the business objective .
- Deployment - After taking our evaluation results, we determine a strategy for their deployment.

3.3 System Architecture

3.3.1 Subsystem decomposition

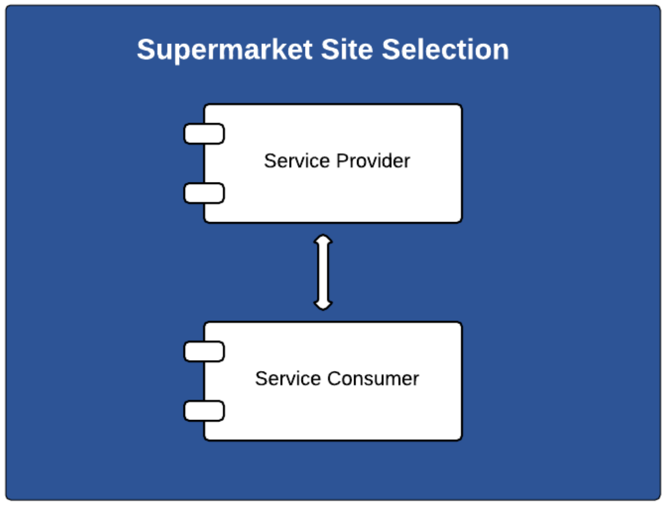

Figure 10 Level 0 Decomposition

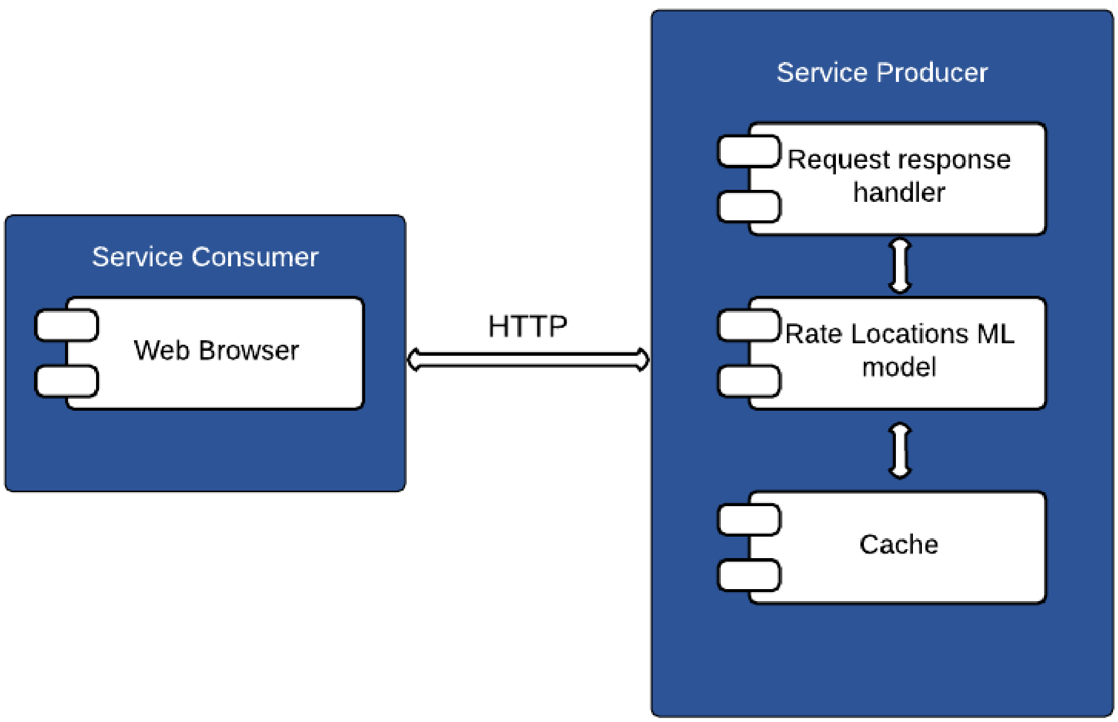

Figure 11 . Level 1 Decomposition

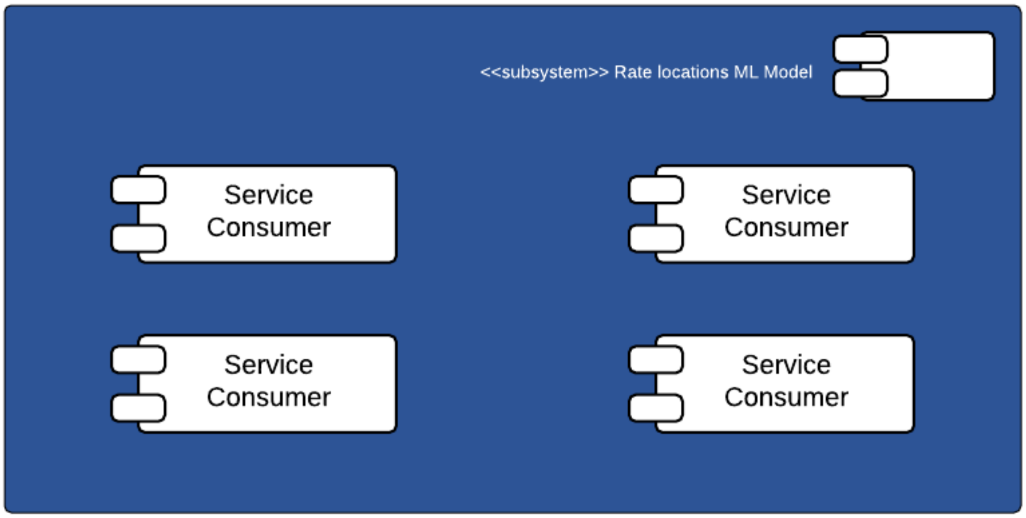

Figure 12 Level 2 Decomposition

3.3.2 Hardware/software mapping

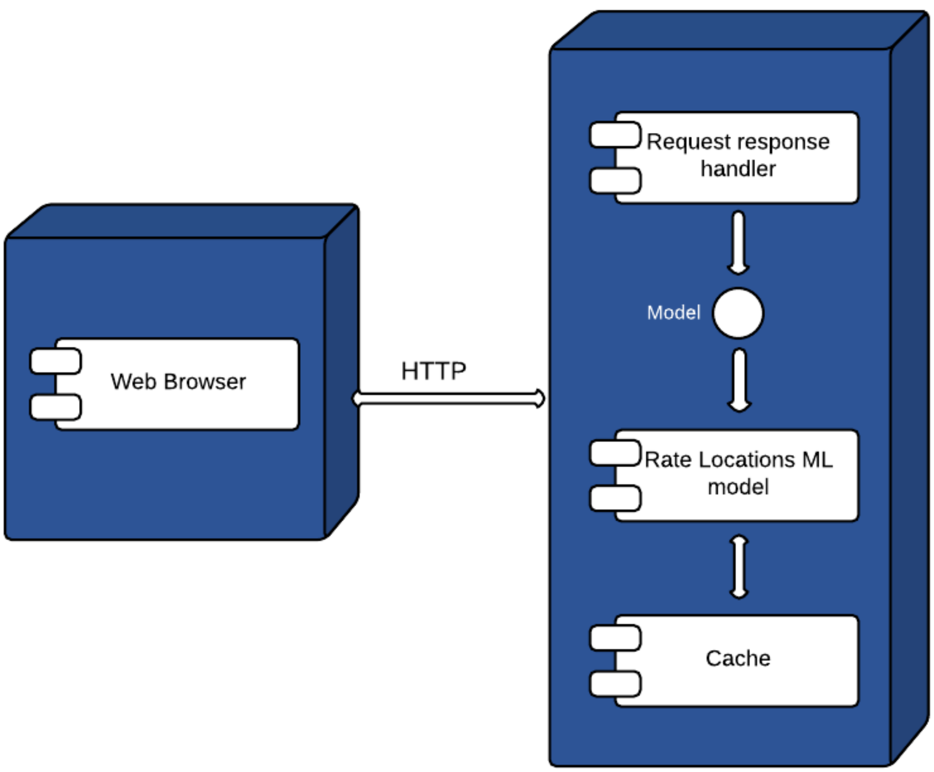

Figure 13 Architecture

3.4 Object Model

3.4.1 Class Diagram

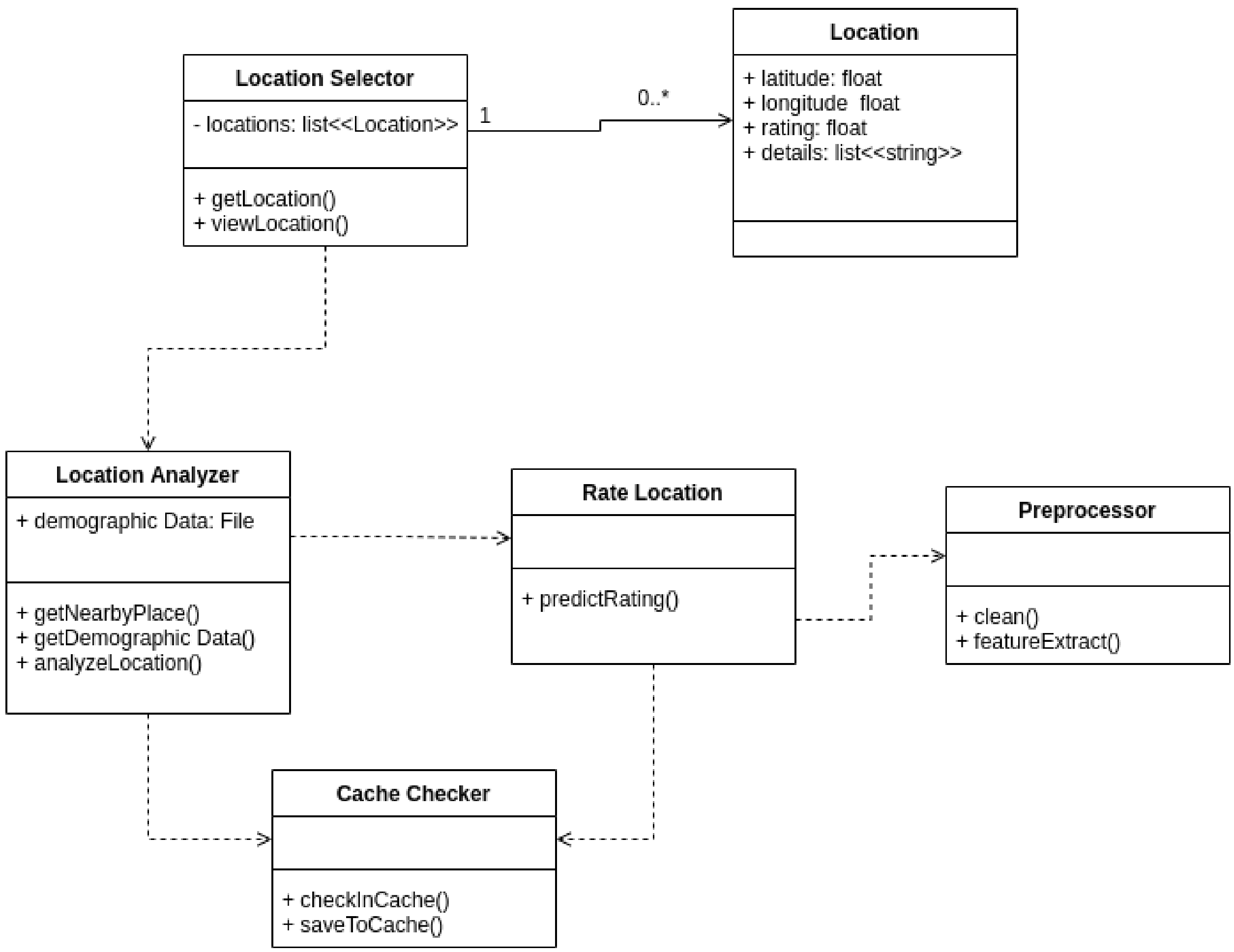

Figure 14 Class Diagram

3.4.2 Sequence Diagram

3.4.3 Select Location

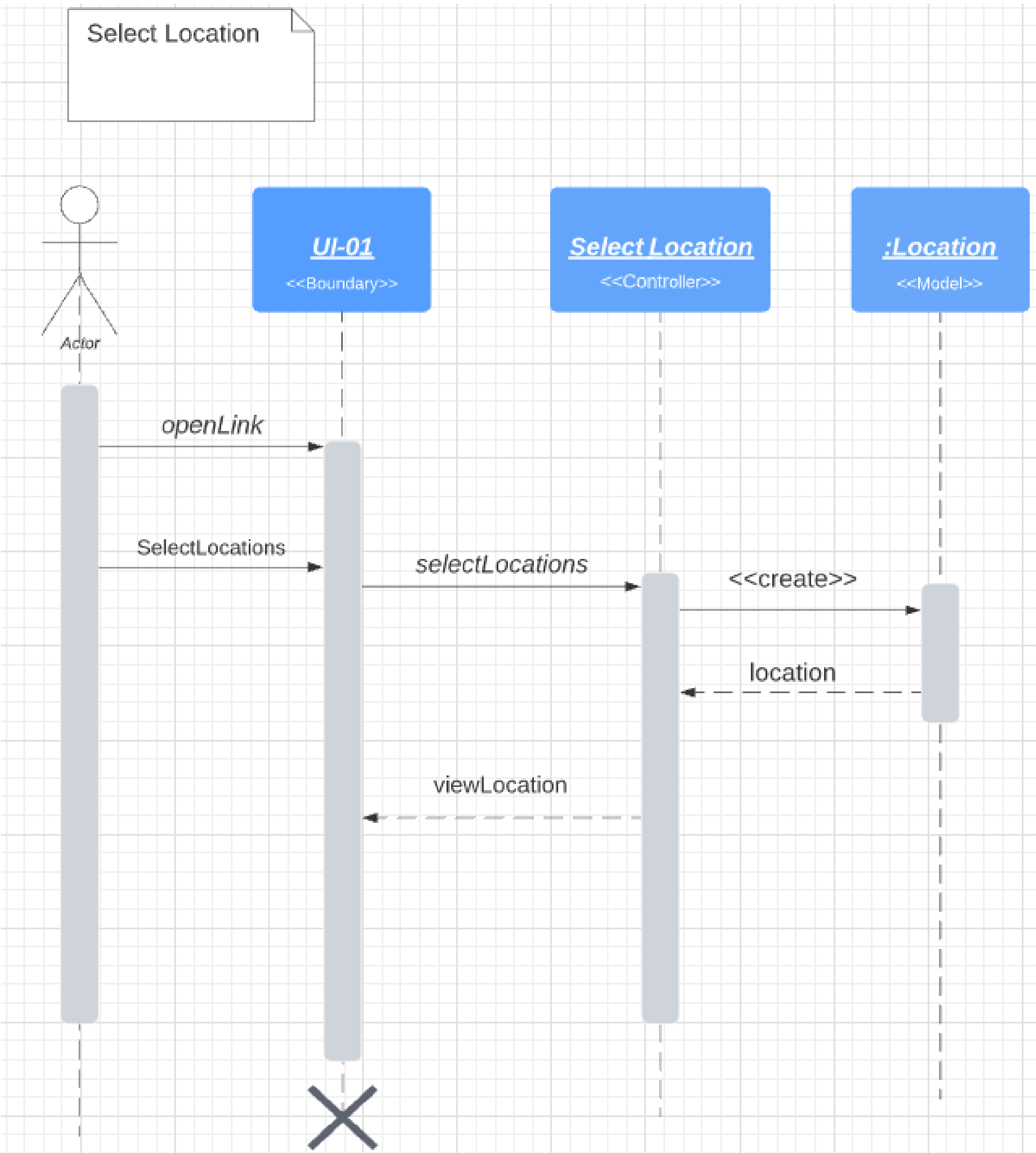

Figure 15 Select Location

3.4.4 Analyze Location

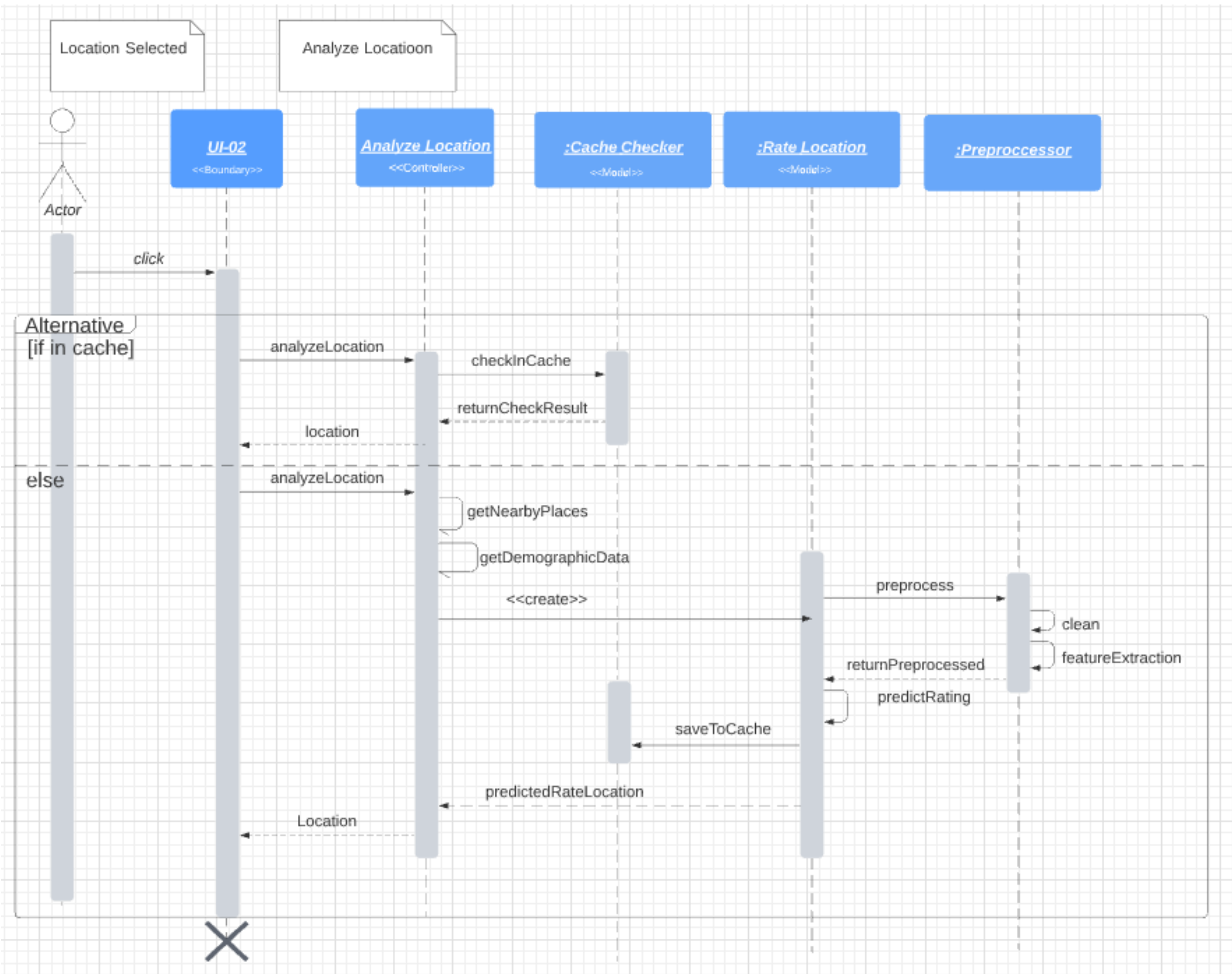

Figure 16 Analyze Location

3.4.5 View Details

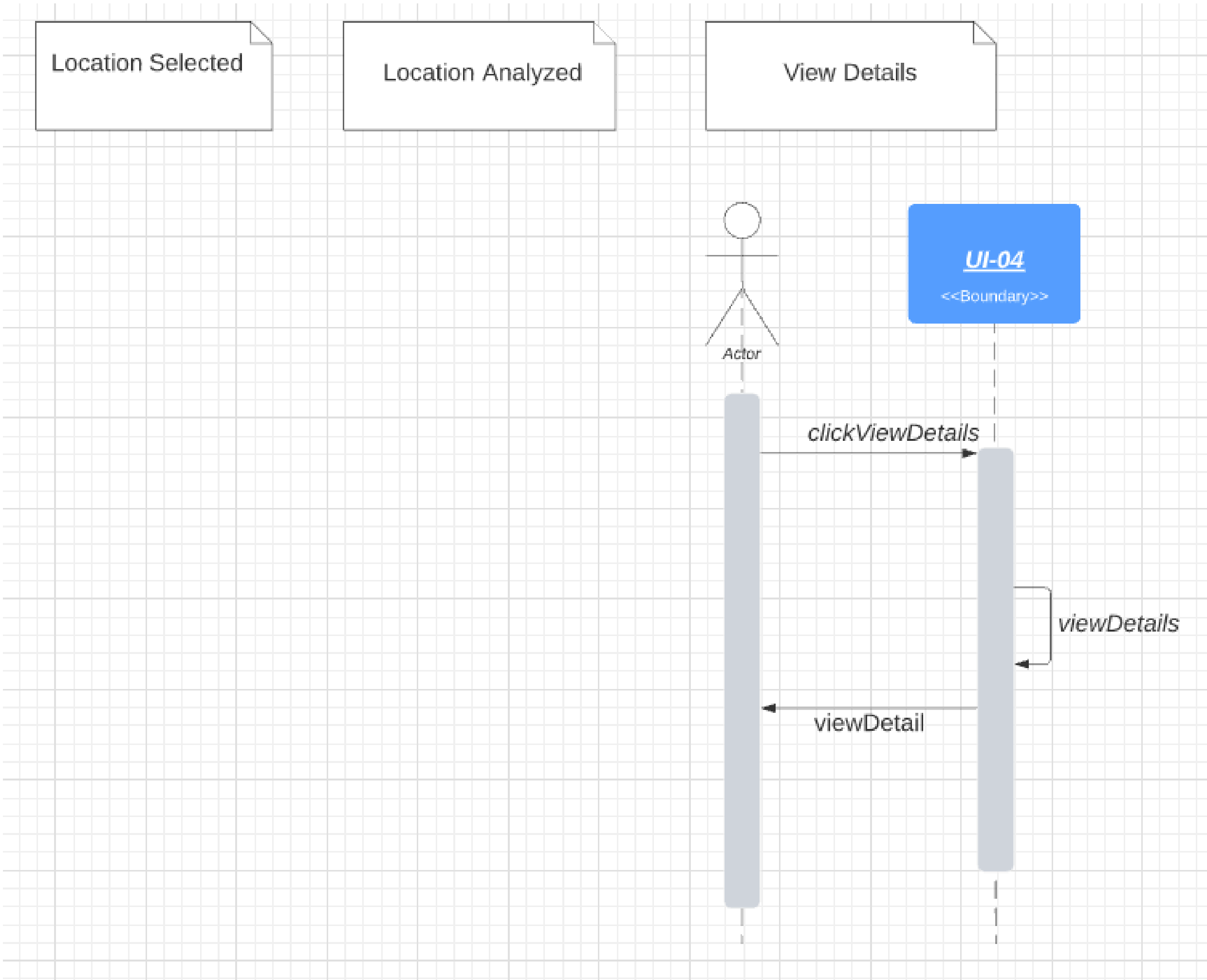

Figure 17 View Details

3.5 Detailed Design

3.5.1 Location Class

Table 16 Location Class Design

Location
+latitude:float +longitude:float +rating:float +details:list<<String>>

Table 17 Attribute Description For Location Class

Attribute	Type	Visibility	Invariant
latitude	float	Public	latitude<>NULL
longitude	float	Public	longitude<>NULL
rating	float	Public	-
details	list<<String>>	Public	-

3.5.2 Location Selector Class

Table 18 Location Selector Class Design

Location Selector
-locations:list<<Location>>
+getLocation () +viewLocation()

Table 19 Attribute Description for Location Selector Class

Attribute	Type	Visibility	Invariant
locations	Location	Private	Location<> NULL. Must contain longitude and latitude and an optional value of the rating, details of the location.

Table 20 Operational Description for Location Selector Class

Operation	Visibility	Return type	Argument	Pre-Condition	Post Condition
getLocation	Public	void	Location	The user must select the location by clicking on the map	Map has loaded
viewLocation	Public	void	Location	The user must select a location	The system returns the locations selected and views them on the sidebar.

3.5.3 Rate Location Class

Table 21 Rate Location Class Design

Rate Location
+predictRate() -preprocessor()

Table 22 Operational Description for Rate Location Class

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
predictRate	Public	List	locations, cached results	Location data is preprocessed	System Predicts and assigns ratings for the locations
preprocessor	Private	void	locations	Location data is organized into train, validation and test set	The system performs feature engineering and normalizes location data.

3.5.4 Location Analyzer Class

Table 23 Location Analyzer Class Design

Location Analyzer Class
+demographicData: File
+getNearbyPlace () +getDemographicData () +analyzeLocation()

Table 24 Attribute Description For Location Analyzer Class

Attribute	Type	Visibility	Invariant
demographicData	File	Public	File<> NULL

Table 25 Operational Description For Location Analyzer Class

Operation	Visibility	Return type	Argument	Pre-Condition	Post Condition
getNearbyPlace	Public	List<<string>>	location	The user must select the location and hit the analyze button	The system retrieves the nearby places of the specific location
getDemographicData	Public	List<<string>>	location	The user must select the location and hit the analyze button	The system retrieves the demographic data of the specific location
analyzeLocation	Public	List<<string>>	List	A location's demographic data and nearby places are retrieved	The system analyzes the data retrieved from a specific location.

3.5.5 Preprocessor Class

Table 26 Preprocessor Class Design

Preprocessor
+clean() +featureEngineer()

Table 27 Operational Description for Preprocessor Class

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
clean	Public	List	List<<string>>	Location data is selected	System removes unnecessary information and cleans data
featureEngineer	Public	List	List<<string>>	Location data is cleaned	The system performs feature engineering and normalizes location data.

3.5.6 Cache Checker Class

Table 28 Cache Class Design

Cache Checker
+checkInCache() +saveToCache()

Table 29 Operational Description for Cache Class

Operation	Visibility	Return type	Argument	Pre-Condition	Post-Condition
checkInCache	Public	List<<string>>	location	The user must select the location and hit the analyze button	The system checks if a nearby location was already predicted for and returns if existent.
saveToCache	Private	void	List<<string>>	Location rating is predicted	Results are cached into the system

3.6 Pseudocode and Algorithm Analysis

3.6.1 Data collection

The dataset is a combination of nearby places and demographic data for each supermarket. Here are the steps to collect the data:

i. **Locate supermarkets**

ii. **Gather nearby places:** gathering data such as the number of ATMs, the number of Bus stations, the number of supermarkets, etc for each supermarket within 500m radius. We chose 500m because it is proven to work on paper[2], as it is not very large and at the same time not too small.

The number of objects fetched using Google Places API serves as information about the structure of potential customers for a given area. For instance, the number of schools and universities is a proxy variable for educational structure of population, whereas, the numbers of bus stations, gas stations, subway stations and taxi stands are proxies for transportation structure of the given area. Some variables, such as, the number of casinos and the number of night clubs and pubs give an idea of customer population in certain time intervals[2].

Items		
accounting	embassy	museum ✓
airport	establishment	night_club
amusement_park	finance	painter
aquarium	fire_station	park
art_gallery	florist	parking ✓
atm ✓	food ✓	pet_store
bakery	funeral_home	pharmacy ✓
bank	furniture_store	physiotherapist
bar ✓	gas_station	place_of_worship
beauty_salon	general_contractor	plumber
bicycle_store	grocery_or_supermarket ✓	police ✓
book_store	gym	post_office
bowling_alley	hair_care	real_estate_agency
bus_station ✓	hardware_store	restaurant ✓
cafe ✓	health ✓	roofing_contractor
campground	hindu_temple	rv_park
car_dealer	home_goods_store	school ✓
car_rental	hospital ✓	shoe_store
car_repair	insurance_agency	shopping_mall ✓

Figure 18 List of data we collect from Google Maps API

iii. Collect demographic data per Woreda: *The main features are population, the number of male and females by age group, and the number of households.*

Figure 19 demographic data age distribution

Table 30 Demographic Data Example

SubCity id	Male population	Female population
1	2341	5424

iv. Combine the gathered data

Table 31 Combined Data Example

ID	Restaurants	Supermarkets	Hospitals	Schools	Banks	Religious Institutions	Gas Stations	ATMs
01	4	7	3.	2	6	2	1	5

Bus Stops	Woreda	Rating
10	9	4

3.7 Train the model

i. Organize the data into train, validation and test set

ii. Load the data

iii. Preprocess

- Read data
- Perform feature engineering

    i. Selecting the best features and creating new features by combining them. For example, from Google nearby we have shopping mall and grocery_or_supermarket as different attributes. By combining these two we can say both as supermarkets.

```

shopping_mall = get('shopping mall')
grocery_or_supermarket  = get('grocery_or_supermarket ')
supermarket = []
for i in range(len(shopping_mall)):
    supermarket[i] = average(shopping_mall[i],grocery_or_supermarket[i])

```

Figure 20 pseudocode for feature engineering

- Normalize the data: The goal is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values to get local minimum fast.

iv. Train the model

- k-means algorithm: for clustering the data by rating. k-means works by calculating the distance

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \left| x - m_i \right|^2$$

- ANN once we have the annotated data clustered by k-means algorithm we can train our network using ANN

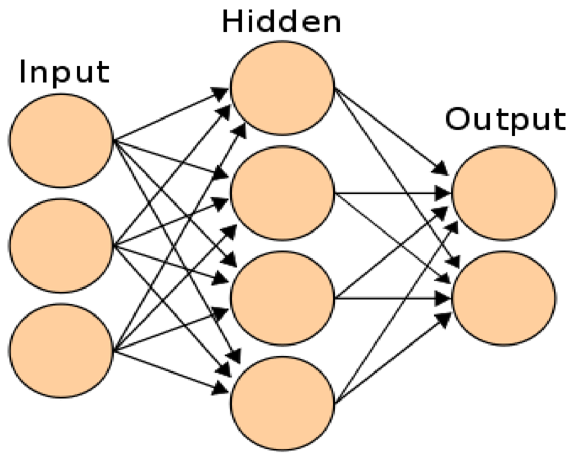

Figure 21 Neural Networks Design

Table 32 Neural Network Details

Parameter	**Value**	**Description**
Inputs	11	Selected by best performance through feature engineering
Hidden Layers	1	120 Neurons with Sigmoid transfer function
Output	6	Bit vectors for Rating. this is predicted rating of the model which rates from 0-5

v. Validate the model

vi. If validation can be improved, go to step 1

3.8 Prediction

i. Preprocess the data

ii. Call "predict method" on the trained model

4. Chapter 4: Testing

4.1 Introduction

This document is the Master Test Plan for Location Intelligence: Supermarket site selection. The standard template adopted is the **IEEE-829 Test plan document template** . The objective of this document is primarily to test:

- Correctness of the implementation of the features described in SRS
- Correctness of Module interfaces and interactions with respect to those described in the SDS.

The project will have three levels of testing; unit level testing, system/integration testing, and acceptance testing. Furthermore a separate Test Design Document has been prepared in order to give detail on test cases

4.2 Features to be tested/not to be tested

4.2.1 Features to be tested

The following is a list of the areas to be focused on during testing of the location intelligence platform.

Table 33 Features to be tested

Feature	**Description**	**Level of Risk**
Select Preferred Location. [FR1]	This feature allows the user to select preferred locations by navigating the map.	High
Analyze Locations [FR2]	The system analyzes selected coordinates by gathering information from nearby places and demographic data.	High
View Location Details [FR4]	The system shall allow the user to see a location's detail.	Medium

-

4.2.2 Features not to be tested

The following is a list of the areas that will not be specifically addressed during testing of the application

- Rate Locations [FR3]
    - The *Rate Location* feature consists of the *Rating ML Model*. The Model's testing set will not be validated due to its dependency on google maps. The validity of the data is handled by the third party.

4.3 Pass/Fail criteria

The pass or fail criteria for a test item are determined by the sort of testing conducted on it.

- **Unit Test**
  The pass/fail criteria for unit tests are simple, and unittest in Python will decide them. After the unit test suite has finished running, unittest will label each individual unit test as succeeded or failed based on our assert assertions.Once all test cases pass successfully, we will consider our unit testing to be complete.

- **Acceptance Testing**
  Each of our requirements from our Requirements Document will be marked as "not accepted" for acceptance testing. Once the project is completed, the application for a specific need will be designated as "approved" if that requirement has been successfully executed. The development team will also follow this procedure for earlier testing.

- **Load and stress Testing**
  The test item will pass if the application performs to an acceptable level after being loaded to a given capacity. The test item, on the other hand, will fail if the application performs poorly.

4.4 Approach/Strategy

The testing levels for this project consists of Unit Level Testing, System/Integration Testing, and Acceptance Testing.

4.5 Testing Levels

4.5.1 Unit Testing

The whitebox unit testing strategy is applied in two sectors, with the aim of ensuring that the requirements are properly satisfied by the application.

- Front-end Unit Testing
  - Cypress unit testing tool is used to test each of the front-end components in isolation.
- Back-end Unit Testing
  - PyTest framework is used to perform functional testing.

4.5.2 Integration Testing

- Since Integration tests have no strict rule as unit testing, the general direction taken is using PyTest to test functionality , reliability and performance of the API.

4.5.3 System Testing

- The system will be manually tested by the development team.

4.5.4 Testing Types

- The tests are based on the functional and non-functional requirements specified on the requirements specification document.

4.5.5 Testing Methods

- White box testing.
- Manual : System level testing will be conducted manually by the development team.

Test cases with specifications

Table 1: Test case specification for select preferred location

Name: Select Preferred Location				
Purpose: to verify appropriate location is selected				
Test Data= longitude and latitude of a place				
Input	**Expected result**	**Data**	**Actual output**	**Pass/fail**
longitude and latitude inside Addis Ababa region	display selected location	Any longitude and latitude inside Addis Ababa region		
longitude and latitude outside Addis Ababa region	"location is outside Addis Ababa" error is displayed	Any longitude and latitude outsideside Addis Ababa region		

Table 1: Test case specification for Analyze Locations

Name: Analyze Locations				
Purpose: to verify the core functionality , which analyzes a location and gives the rating of a place.				
Test Data= longitude and latitude of a place				
Input	**Expected result**	**Data**	**Actual output**	**Pass/fail**
one longitude and latitude	display rated location	one longitude and latitude inside Addis Ababa region		
multiple longitude and latitude	display rated location	multiple longitude and latitude inside Addis Ababa region		
empty longitude and latitude	"select location" error is displayed	empty longitude and latitude inside Addis Ababa region		

Table 1: Test case specification for Login

Name:View Location Details				
Purpose: to verify users can view details of a location				
Test Data= entry/record ID				
Input	**Expected result**	**Data**	**Actual output**	**Pass/fail**
The unique ID of entry	display Location Details	Any unique ID of entry that is to be viewed		

5. Chapter 5: User Manual

5.1 Scope

This manual covers how to use supermarket site selection software systems.

The manual covers the following:

- How to start the system
- How to select preferred location
- How to view details of analyzed location

5.2 Installation and configuration

The system is web based and doesn't require any special configuration or installation but we recommend the user to use known browsers such as chrome, firefox, safari or microsoft edge.

5.3 How to Operate the system

A. How to start the system

Open the system through the url and it will show a map interface

Figure 22 Landing page of the Application

B. How to select preferred location

There are two ways you can select your preferred location:

i. Enter name of your preferred location in the search box

Figure 23 preferred location through search box

ii. pan around on the map and use left click to pinpoint your preferred location

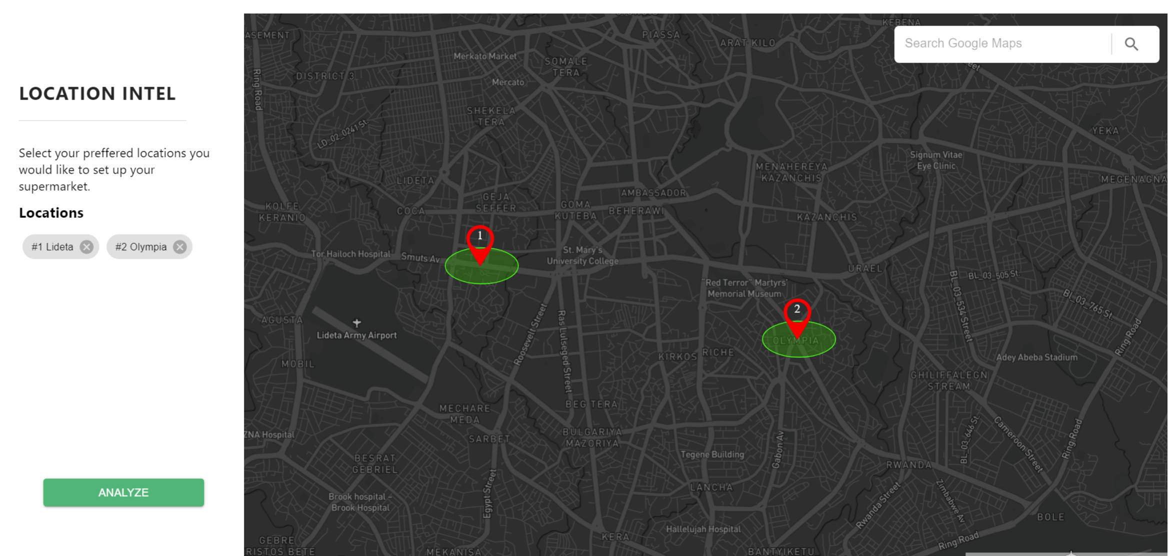

Figure 24 preferred location through panning

C. How to analyze location

Once you select the preferred location click on analyze button

Figure 25 Analyze Locations

D. How to view details of the analyzed location

Once the system analyzes the locations, pan on the map and click the location you wish to view its details.

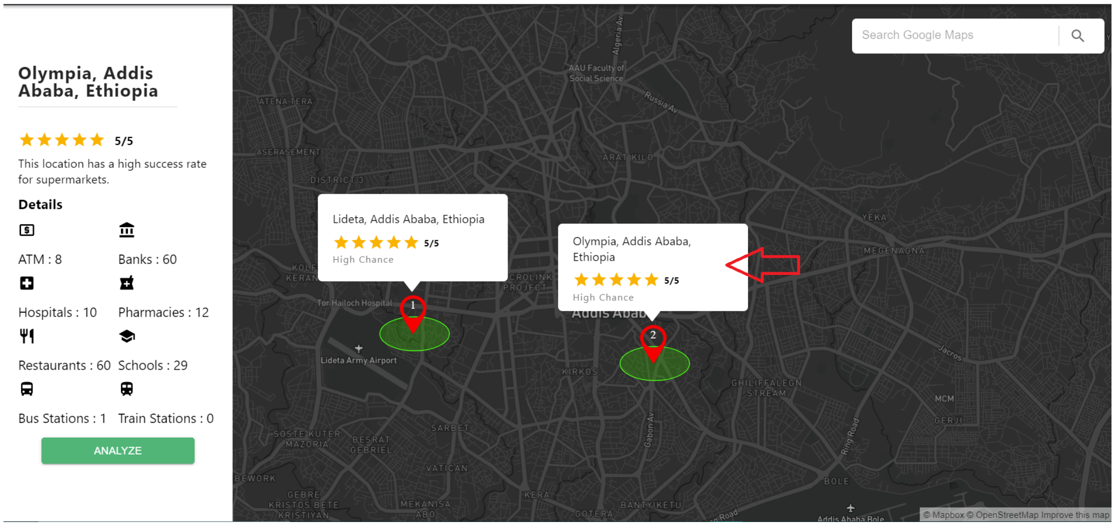

Figure 26 View details of analyzed locations

6. Chapter 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Using Location intelligence, one can get insights quicker, in a more intuitive and cost efficient manner. To apply this technology though, we would need to gather as much data as we can for better accuracy. Other developed countries have the benefit of already built structures and data acquisition practices in play for years from foot traffic data, wealth distribution, business profit projections to many types of data that is enabling them to prosper and see patterns humans cannot see. For developing countries like Ethiopia, acquiring such data is hard despite the lack of data. We have tried to show what we can achieve using the available information around us. We used Google Maps and Census data alongside Machine Learning to showcase the potential of Location Intelligence.

6.2 Recommendation

This project is meant to showcase how location intelligence can be used for businesses. Even Though the system is performing as expected it can be further improved by adding datas like, actual profit of each supermarket instead of relying solely on biased user rating, and expand the range of our demographic data to income and growth of the neighborhood. As this project requires intensive data gathering and years of research to have a real world working system we can only lead the way and inspire future researchers.

7. BIBLIOGRAPHY

[1] *Supermarket Key Attributes and Location Decisions*

[2] *Ethiopia Economic Outlook*

[3] *Underlying Causes of Micro and Small Business Failures in Addis Ketema Sub City*

[4] *Targomo*

[5] *Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API*

[6] *Azure Maps*

[7] *Addis Ababa statics*

REFERENCES

[1] *Supermarket Key Attributes and Location Decisions*

[2] *Ethiopia Economic Outlook*

[3] *Underlying Causes of Micro and Small Business Failures in Addis Ketema Sub City*

[4] *Targomo*

[5] *Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API*

[6] *Azure Maps*

[7] *Addis Ababa statics*