**2023-2024 SPRING SEMESTER**

**SE112 | INTRODUCTION TO SOFTWARE ENGINEERING**

**WHEELCHAIR GUIDANCE & ASSISTANCE APP**

GROUP NO: 21

CEMRE CEVHER YILDIZ - 22244710038

PELİN IŞIK - 23244710114

AYBARS PAKKAN - 23244710016

Table of Contents

**PROJECT PROPOSAL3**

**PROJECT MANAGEMENT**

PROJECT PLAN4

GANTT DIAGRAM5

RISK MANAGEMENT PLAN……………………………………………………………………………………6

**MOSCOW SCHEME**…………………………………………………………………………………………….…**7**

**FUNCTIONAL REQUIREMENTS**

USER STORY……………………………………………………………………………………………………8

USE-CASE DIAGRAM………………………………………………………………………………………..9

i. PLANTUML CODE………………………………………………………………………………………10

SEQUENCE DIAGRAM…………………………………………………………………………………….11

ii.PLANTUML CODE……………………………………………………………………………………..12

CLASS DIAGRAM……………………………………………………………………………………………13

iii.PLANTUML CODE…………………………………………………………………………………14-15

**FIGMA**…………………………………………………………………………………………………………**16-19**

**Wheelchair Guidance and Assistance App**

***Project/Problem Description****:* The aim of the project is to help people with disabilities. It is a mobile application, designed specifically for people who use wheelchairs. The main purpose of this app is to make it possible for them to navigate around freely and determine an environment that makes it easier to travel with a wheelchair. With the features of this app, users can see elevators, ramps on the road, paths that are available for them to take. They can also search for restaurants, cafes, and even apartments to see if they are wheelchair- friendly. Our app can also detect the small obstacles (such as holes, stones etc.) on the road with the help of the sensors and warn the user about them. Additionally, the app offers AI assistance, where users can contact and ask for help during an emergency or if they have any issues they have encountered. On top of that, users can also save the locations they searched to check them out later on. With the help of these features, people who use wheelchairs will be safer, more comfortable, and at ease while living their daily lives.

**PROJECT MANAGEMENT**

**Project Plan**

**Planning**

**-**Defining project objectives

-Establishing the timeline

1. **Gathering Data**

**-**Gathering the required data

-Taking feedback from the potential users

1. **Designing**

**-**Planning the user interface

-Planning the database structure

1. **Implementation**

**-**Writing code

1. **Testing and Integration**

**-**Bringing the software and system components together

-Testing the software

1. **Maintenance**

**-**Maintaining the software after it is launched

-Fixing any possible problem.

-Addressing user request

**GANTT DIAGRAM**

**A graph with a green line

Description automatically generated**

**RISK MANAGEMENT PLAN**

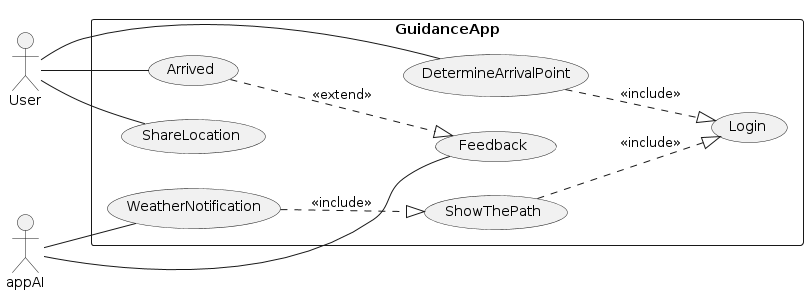
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Description** | **Probability** | **Severity** | **Actions to Minimize Risk** |
| **1)** | | | | |
| Financial problems | Identifying every obstacle, ramp, elevator etc. in a large region is a costly job. We need expensive equipment to scan the region, let alone our staff’s salary and our other expenses. | Certain | Major | We can inform the customer about extra expenses. We can start a GoFundMe campaign. As a last option, we can limit our other expenses to balance out the budget. |
| **2)** |  |  |  |  |
| Requirements change | Requirements may need to be updated after the project initiation. Changing requirements might result from evolving needs, regulatory changes or market dynamics. | Certain | Major | We must assess the impact of the change on our ongoing project. We should evaluate different options and update our project plans, schedules and budget according to the most suitable solution. |

|  |  |  |
| --- | --- | --- |
| **Functional Requirements ID** | **Explanation** | **MOSCOW Priority (M, S, C, W)** |
| FR1 | The app must have a “User Registration” page for users to register. | M |
| FR2 | Users must create a username and a password to register the app. | M |
| FR3 | The system must send a confirmation email to verify user accounts. | S |
| FR4 | The app should send a notification to inform any updates made in the app. | S |
| FR5 | The background colour for the app interface is a soft green. | C |
| FR6 | The system must allow the users to submit feedback. | W |
| FR7 | The app must inform the user about rain/storm situation. | W |
| FR8 | The system should encrypt sensitive user data to ensure security. | M |
| FR9 | Provide error messages when users provide invalid or incomplete inputs. | C |
| FR10 | Users should be able to share their current locations with other users. | W |
| **Non-Functional Requirements ID** | **Explanation** | **MOSCOW Priority (M, S, C, W)** |
| NFR1 | The system must support 20.000 users simultaneously trying to use the app. | S |
| NFR2 | The processing of each request should be done within 5 seconds. | S |
| NFR3 | Passwords must be at least 8 characters long for user authentication. | C |

**User Story**

Wheelchair Guidance and Assistance App is used to help disabled people travel safely. For example, after the users create their accounts, they determine their arrival point. The app shows them the most suitable path options. App AI also sends a notification about the weather condition at that moment, so they can be prepared. Then, the users choose if they want to share their location with other users. After they arrived at their destination, they must inform the system that they arrived. Then, they will be asked to give feedback by our AI about their journey.

**USE CASE DIAGRAM**



**PLANTUML CODE**

@startuml

actor User

actor appAI

'top to bottom direction

left to right direction

rectangle GuidanceApp{

usecase DetermineArrivalPoint

usecase Login

usecase ShowThePath

usecase WeatherNotification

usecase ShareLocation

usecase Arrived

usecase Feedback

}

User -- (DetermineArrivalPoint)

(User)-- (ShareLocation)

(User) -- (Arrived)

(appAI) -- (WeatherNotification)

(appAI) -- (Feedback)

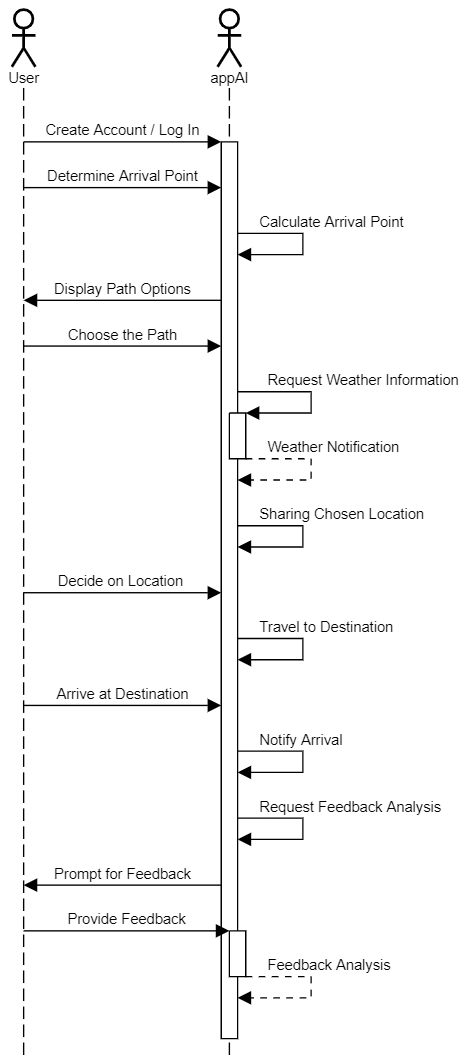
(DetermineArrivalPoint) ..|> (Login) : <<include>>

(ShowThePath) ..|> (Login) : <<include>>

(WeatherNotification) ..|> (ShowThePath) : <<include>>

(Arrived) ..|> (Feedback): <<extend>>

@enduml



**SEQUENCE DIAGRAM**

**PLANTUML CODE**

actor User

actor appAI

User -> appAI: Create Account / Log In

activate appAI

User -> appAI: Determine Arrival Point

appAI -> appAI: Calculate Arrival Point

appAI -> User: Display Path Options

User -> appAI: Choose the Path

appAI -> appAI: Request Weather Information

activate appAI

appAI --> appAI: Weather Notification

deactivate appAI

appAI -> appAI: Sharing Chosen Location

User -> appAI: Decide on Location

appAI -> appAI: Travel to Destination

User -> appAI: Arrive at Destination

appAI -> appAI: Notify Arrival

appAI -> appAI: Request Feedback Analysis

appAI -> User: Prompt for Feedback

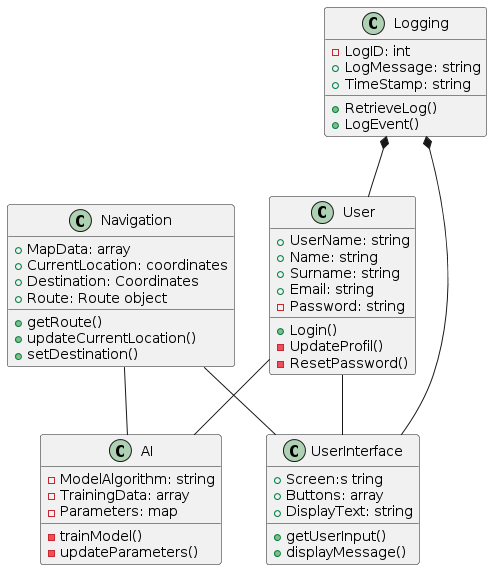
User -> appAI: Provide Feedback

activate appAI

appAI --> appAI: Feedback Analysis

deactivate appAI

**CLASS DIAGRAM**

****

**CLASS DIAGRAM PLANTUML CODE**

@startuml

class User{

+UserName: string

+Name: string

+Surname: string

+Email: string

-Password: string

+Login()

-UpdateProfil()

-ResetPassword()

}

class Logging{

-LogID: int

+LogMessage: string

+TimeStamp: string

+RetrieveLog()

+LogEvent()

}

class Navigation{

+MapData: array

+CurrentLocation: coordinates

+Destination: Coordinates

+Route: Route object

+getRoute()

+updateCurrentLocation()

+setDestination()

}

class AI{

-ModelAlgorithm: string

-TrainingData: array

-Parameters: map

-trainModel()

-updateParameters()

}

class UserInterface{

+Screen:s tring

+Buttons: array

+DisplayText: string

+getUserInput()

+displayMessage()

}

User -- UserInterface

User -- AI

Navigation -- UserInterface

Navigation -- AI

Logging \*-- User

Logging \*-- UserInterface

@enduml

**A map with a pin on it

Description automatically generated**

**A screenshot of a phone

Description automatically generated**

A screenshot of a login form

Description automatically generated

**A screenshot of a phone call

Description automatically generated**