EGE UNIVERSITY

COMPUTER ENGINEERING

- **Public Transport Route Finder System (PTRFS)**

REQUIREMENTS ANALYSIS REPORT (19.11.2018)

- **Project Members:**

  - **05140000695 - Burak ÖZKAN**
  - **05140000680 - Emin Soner TÜRK**
  - **05140000632 - Furkan DEMİR**
  - **05170000019 - Ramazan ONBAŞI**

Contents

1)Introduction

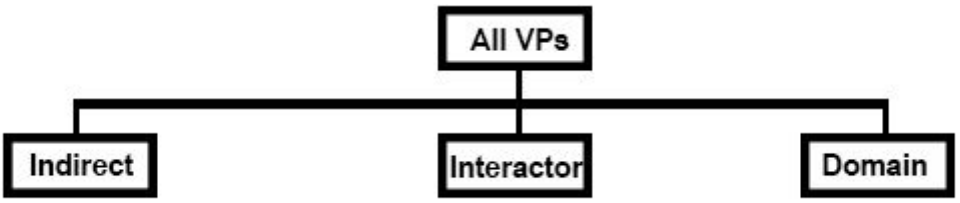
Public Transport Route Finder (PTRF) is a mobile application with an active Internet connection, which can be used by people living in many different cities all over the world. Main aim of the application is to suggest the available public transport routes between two given points in a city. Depending on the means of public transport available in a city, it can suggest different kinds of transport types including railway, land vehicles (city busses, private buses, etc.) and sea transport. If the stations (bus stops, etc.) of two consecutive vehicles to be used are within walking distance, the system should also show the shortest path to be walked to reach the next station on the route. It should also be possible to add new public transport types and update the existing ones in each city.

In this system analysis report we want to show the requirements, actors, stakeholders and viewpoints of the system so that we can see the small parts of a big structure. The structure itself will become easier to understand when we disintegrate it into smaller parts.

2)Identification of Viewpoints

The viewpoint-oriented software development defines the related roles as viewpoints. The actor of the requirements become a viewpoint to the system and their need is defines the path of the developer. There are three generic viewpoints to discover.

- Interactor viewpoint: People or other systems interact directly with the system.
- Indirect viewpoints: Stakeholders who don’t use system directly but have some requirements.
- Domain viewpoint: Domain properties that influence the requirements.



Principle Viewpoints of the System:

- Viewpoints for the Public Transport Route Finder System (PTRFS) system.

Interactor Viewpoints:

- Passenger
- Uses the program to find which public transports he/she needs to take in order to go from point A to point B. Also the passenger can give feedbacks about the program.

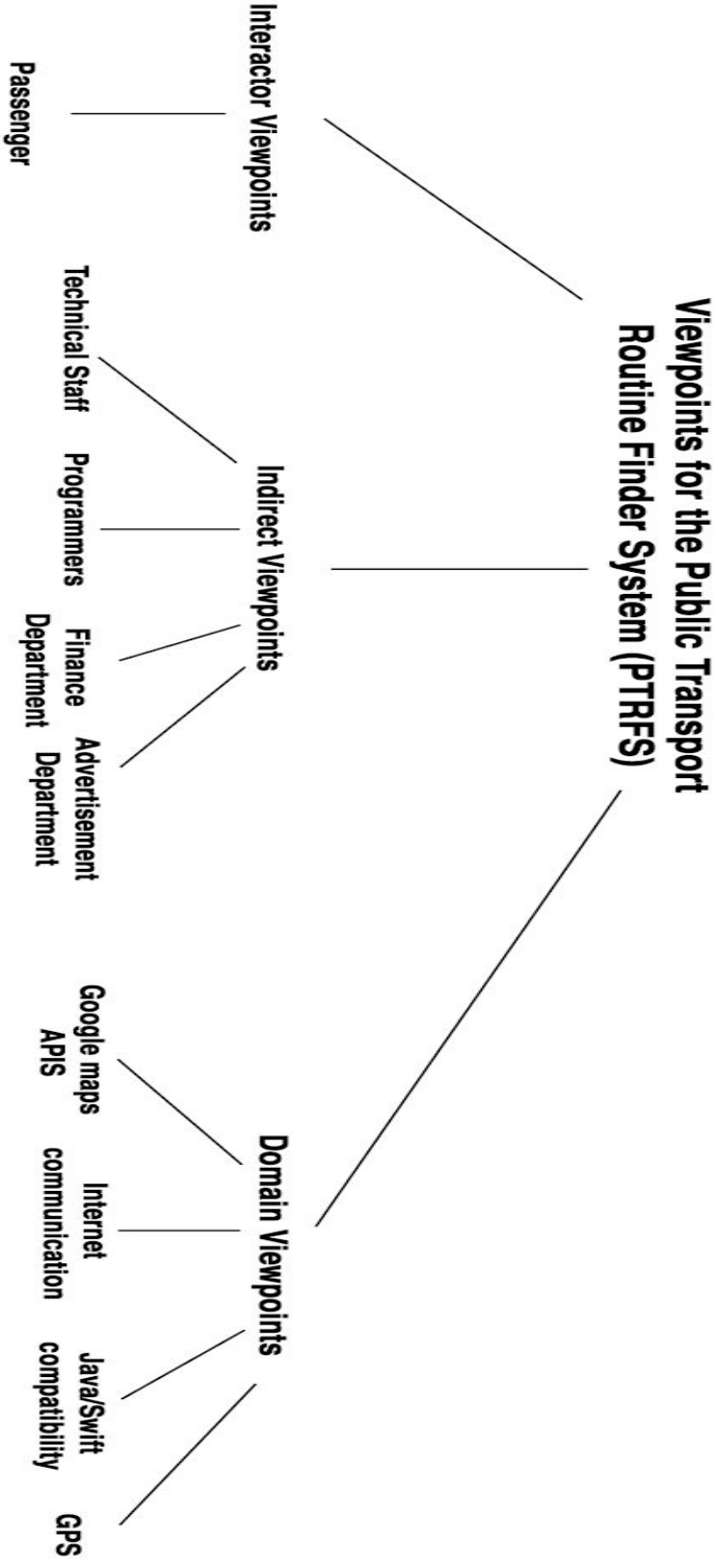
Indirect Viewpoints:

- Technical Staff
    - Technical team who installs local servers and equipments.
    - Technical team who collects data (e.g. bus departure times) and updates the program with the aforementioned data.
- Programmers
    - Programmers who maintain and further develop the program according to the feedbacks and new/changed requirements.
- Finance Department:
    - Manages balances.
- Advertisement Department:
    - Advertises the program.

Domain Viewpoints

- Google maps API
- Internet communication
- Java/Swift compatibility
- GPS

Viewpoint Hierarchy Diagram



Requirements of each Viewpoint

R1. Passenger: Passenger wants to know how to go point B from point A by using public transports.

R2. Technical staff: Technical stuff wants to monitor the hardware and software problem of the system. Also they want to be able to update the information of the system.

R3. Programmers: Programmers want to access the source code in order to maintain and improve the program.

R4. Finance Department: Finance department wants to make a profit.

R5. Advertisement Department: Advertises the program in order to increase its popularity.

R6. Domain: It wants to provide the necessary environments for the program.

3)Requirements Definition(Considering Functionality)

Functional Requirements:

1. System shall show the user how to go from point A to point B.
2. System should be able to take the coordination of the user and should show user's location on the map.
3. System should show the map on the screen.
4. System should present the road conditions to the users.
5. System shall allow technical staff to add/delete public transport instances.
6. System shall allow technical staff to add new kinds of public transports.
7. System shall allow technical staff to add/delete/update local special rules (such as 90 minute rule in İzmir) for public transports.
8. System shall allow technical staff to update the information regarding the routes of public transports.
9. System shall have 3 different algorithms to select from. One of them shall find the shortest route, another one shall find the fastest route (i.e. it shall take traffic into account), and the last one shall find what is the cheapest way to get there (i.e. it shall take the special rules, such as 90 minute rule, into account).
10. System shall allow users to select maximum walking distance.
11. System shall allow users to select what kind of public transports they are comfortable/uncomfortable with.
12. System shall allow users to give feedback about the program itself and about how accurate the information is for a specific public transport.
13. System should provide multiple languages.
14. System should support three kinds of accounts: administrator, registered user and guest account.
15. System should allow users to download additional packages for other cities/countries. By default the program will only have the package for the local which the user currently resides in.
16. System shall be able to find the nearest station.

Non-functional Requirements

a. There should be a database system.

b. Response time should be under 5 seconds.

c. The user should be able to learn how to use the application within 5 minutes.

d. System should support Android and iOS.

e. Acceptable failure rate of the system shall be under 1:1000.

f. System should be able to recommend public transports which are suited for people with disabilities.

g. Users should be able to use the system even when they are offline.

h. Between 06.00 and 23.50 in any one day, the total system down time should not exceed 1 hour.

i. System should present at least an alternative public transport.

Domain Requirements

i. Users must accept the terms of service of Google Maps.

ii. The mobile device of the user must support GPS and/or Internet connection.

iii. Users must accept the terms and conditions of iOS and/or Android.

iv. Programmer must accept the terms and conditions of Java and/or Swift.

4)Requirements Definition (Considering Lifetime)

Volatile Requirements

- System should support Android and iOS.
- Acceptable failure rate of the system should be under 1:1000.
- The user should be able to learn how to use the application within 5 minutes.a

Enduring Requirements

- System shall show the user how to go from point A to point B.
- System should be able to take the coordination of the user and should show user's location on the map.
- System should present the road situation to the users and passengers.
- System should allow technical staff to update the information regarding the routes of public transports.
- System should be able to find the nearest station.

5)Requirements Interview with Domain Expert

An Interview with a Bus Driver

1). How often do the passengers ask about the location of bus-stop which the bus they will take stops at?

Answer:

- Approximately 30 out of 100 passengers ask that kind of question per a day.

2). At which time intervals do the traffic get busy?

Answer:

- Between 08.00-09.00 and 17.00-19.00 traffic gets pretty busy.

3). How often do the location of bus stops get changed on route?

Answer:

- Not very often. It depends on how many passengers use that certain bus stop and how often that certain bus get used. But generally speaking I'd say the maximum numbers 2 or 3 times per year at most.

4). If you were to introduced to an application which tells how to go from point A to B would you be willing to give feedbacks?

Answer:

- It depends on the software to be honest. If the said software gets the job done and is popular enough, I guess I'd be willing to give some feedbacks about both the software itself and about environmental changes which can affect the software such as roadworks and whatnot.

5). How often do you see people with disabilities using public transports and how many buses would you say are suited for such people?

Answer:

- I would say I see them using public transports quite often. Well, I can say that I've never seen a bus which isn't suited for people with disabilities.

İlyas TURGUT, 05076285409
ilyasturgut7@gmail.com

6)Requirements Prioritization and Negotiation

6-1)Requirements Prioritization

F: functional requirement
NF: non-functional requirement
D: domain requirement

(Priority, in descending order.)

Functional Requirements

F_1, F_6 , F_5, F_8, F_7, F_9, F_4, F_16, F_10, F_11, F_2, F_12, F_15, F_3, F_13, F_14

Non-functional Requirements

NF_d, NF_a, NF_e, NF_g, NF_b, NF_h, NF_i, NF_c, NF_f

Domain Requirements

D_iv, D_iii, D_ii, D_i

6-2)Requirements Negotiation

- Some roads may not be legal for pedestrians to walk along. But some users may want those roads to be taken into account nonetheless. In order to resolve this conflict, it is decided that the application will show those roads as well as other roads but it will clearly state that walking along those roads is illegal.

- Normally, our application needs to show how to go from point A to point B with public transports that takes the shortest path possible. But shortest path doesn't always mean that it is the fastest or cheapest way there is. Also, some users might want to avoid certain kinds of transports even if it is the cheapest and fastest way to go from point A to B. In order to resolve this conflict, it is decided that the application will let users select which kinds of vehicles they want to avoid and which algorithm they want to use (i.e. shortest, cheapest or fastest route).

- Normally we want to be able to take the current location of the user. But some users may have privacy or security concerns about that kind of information. To resolve this conflict, it is decided that users will be asked whether they want their location to be pinpointed for them. If they don't want to disclose their current location, they can choose a point A manually.

- On one hand, we don't want to allow any operations which might be harmful. On the other hand, in order the system to work properly, we need to allow technical staff to delete/update information regarding vehicle types/vehicles/special rules. While this is necessary, it also creates liabilities. In order to solve this conflict, it is decided that system will keep logs about every operation. And a backup of the system will be taken regularly.

- On one hand, we want to get feedback from users and correct any mishap there is. On the other hand, not all the feedback which come from users will be trustworthy. So we cannot simply commit all the feedback directly into the system. Therefore, it is decided that all the feedback will be checked by technical staff before they are committed into the system itself.

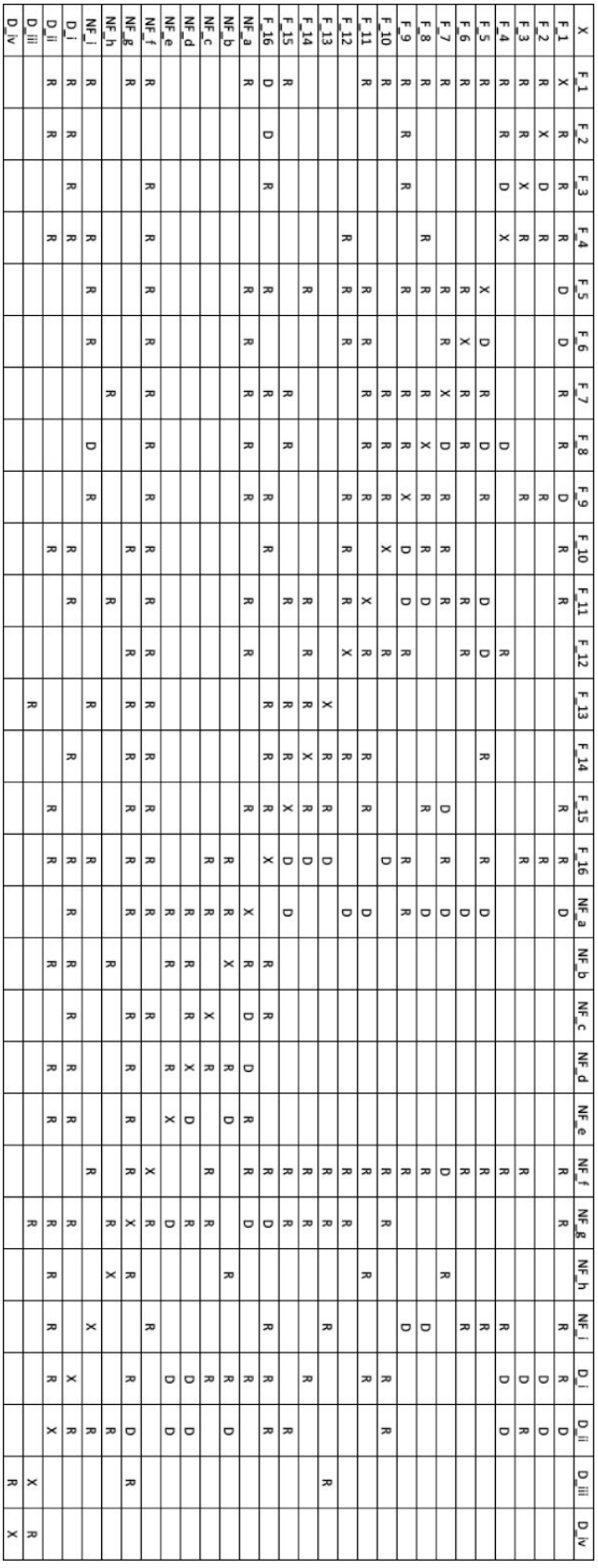
- Normally, users need to sign-up in order to sync their preferences across different devices and to further customize the app. But some users may be too lazy to create an account, or they may think that those features are unnecessary for them. Or they may not want to share any information regarding themselves for whatever reason. To resolve this conflict, it is decided that having a proper account will be optional and users who wish to use the application without a proper account will be able to use it with a guest account.

- Since this application needs to work everywhere in the word, it should normally download all the needed information for the whole globe. But this would take up a huge storage space and chances are that most of the users won't travel the whole world. So it is decided that the information about countries will be divided into separate packages and users will be able to download just the packages that they need.

- Normally, the application needs internet connection to be fully functional and to have up-to-date information. But users may not always have an internet connection. Because of this, it is decided that the application will offer an offline mode as well. When it is used without internet connection, the application will warn the user that the information that application shows may be outdated. Also, when there’s no internet connection, the "fastest route" algorithm will be grayed-out because the application cannot get the needed information regarding the current traffic.

7) Requirements Traceability Matrix

Dependent: D; Related: R; Unrelated: Empty Space;
The graph reads from Y axis to X axis.



X	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_10	F_11	F_12	F_13	F_14	F_15	F_16	NF_a	NF_b	NF_c	NF_d	NF_e	NF_f	NF_g	NF_h	NF_i	D_i	D_ii	D_iii	D_iv
F_1	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	D	R	R	R	R	R	R	R	R	R	R	R	R	R
F_2	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	D	R	R	R	R	R	R	R	R	R	R	R	R	R
F_3	R	R	X	D	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_4	R	R	D	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_5	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_6	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_7	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_8	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_9	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_10	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_11	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_12	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_13	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_14	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_15	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R	R
F_16	D	D	R	X	R	R	R	R	R	D	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R	R
NF_a	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R	R
NF_b	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R	R
NF_c	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R	R	R	R
NF_d	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	D	R	R	R	R	D	R	R	R
NF_e	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	D	X	R	R	R	R	D	R	R	R
NF_f	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R	R
NF_g	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R	R
NF_h	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R	R
NF_i	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R	R
D_i	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X	R	R	R
D_ii	D	D	R	D	D	D	D	D	R	R	R	R	R	R	R	R	R	R	R	D	R	R	D	R	R	R	X	R	R
D_iii	D	D	D	D	D	D	D	D	D	D	D	D	R	D	D	D	D	D	D	D	D	D	D	D	D	D	R	X	R
D_iv	D	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	X

8) Fully Dressed Use Cases of Main Scenarios

Use Case UC1: Road recommendation between two entered locations

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primary Actor: Passenger

Stakeholders and Interests:

- Passenger : Passenger wants to learn how to go from one location to another by using public transports.

Precondition:

- The passenger must turn on GPS and/or should have internet connection. Or he/she can manually enter his/her location.
- The passenger must have entered destination location.

Postconditions:

- The system must be able to successfully navigate according to the entered location information.

Main Success Scenario:

1) If GPS is on, the system can automatically receives the passenger's location information.
2) If GPS is off or if the user does not want to disclose his/her location automatically, the system asks passenger to enter his/her departure location.
2.1) The passenger enters his own location.
3) The passenger selects the location where he/she wants to go from the map.
4) The passenger selects which algorithms will be used.
5) The system provides the user with recommendations of the road.
5.1) If the passenger has entered the input according to the shortest time selection, the program provides directions according to the shortest time.
5.2) If the passenger has entered the cheapest direction selection, the program makes a directions according to the cheapest way.
5.3) If the passenger has selected the fastest transportation algorithm, application will suggest routes considering the traffic as well.
5.4) If the passenger is disabled and wants a recommendation according to him/her, it gives the directions of the access roads with disabled ways and disabled support.
6) The passenger selects one of the recommended routes and gets a detailed instruction on how to he/she can go to the aforementioned destination.

Extensions:

1) If GPS is off,

· The system gives a warning message that GPS is off and prompts manual input of position information.

1) If there's no internet connection,

· System warns the user with saying that the information on the application might be outdated.

5) If system there's no route that fits to user's preferences,

·System will warn the user that there's no route that fits user's settings. And will recommend alternative routes even though they do not exactly fit into user's settings. Those recommended routes may have more walking distance then user originally set or it may be that user might need to take a certain type of transport to get where he/she wants to go, even though user set that kind of vehicle to be avoided.

5) If the system is unavailable for some technical reasons.

· The system will let passenger know that the system is not available at the moment.

Use Case UC2: Register in the system

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Guest

Stakeholders and Interests:

- Guest :Want to become a member of the System.

Precondition: User should have an internet connection.

Postconditions: Guest have been registered into the system.

Main Success Scenario:

1. Guest enters into the registration screen.
2. The system sends a registration form to the guest.
3. Guest will enter the information required for the registration process. Guest clicks the SIGN UP button.
4. The system will display a text which will say "Your registration successful".

Extensions:

3.a) If there are some necessary fields which were left empty,

- System will warn the user that they need to fill those fields out.
- Guests check and refill the missing information and sends this information to the system.

3.b) If there's already a user with the same username or e-mail address,

- System will warn the user that there's already an user with the same info. And will provide a password recovery option in case user forgot his password.
- Guest will either refill the form again if the said account does not belong to him/her, or will try to recover his/her password.

Use Case UC3: Login to system

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Guest

Stakeholders and Interests:

Guest: Guest wants to log in into the system.

Precondition: The user is already registered.

Postconditions: Guest logs in to the system.

Main Success Scenario:

1. Guest views the login screen.
2. The system sends a form for logging in.
3. Guest fills in the form.
4. Guest presses the Login button.
5. Guest is logged into the system.

Extensions:

4.a) If an account with the given information cannot be found

· The system will warn the user that their username/email is incorrect.

· Guest will be redirected to the login screen.

Use Case UC4: Log-out

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: User

Stakeholders and Interests:

User: User wants to log out from the system.

Precondition: User must be logged into the system.

Postconditions: Guest will be logged out from the system.

Main Success Scenario:

1. User presses the log out button.
2. The system terminates the session and logs out from the account.

Use Case UC5: Add new kinds of public transports

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Technical staff

Stakeholders and Interests:

Technical staff: Technical staff wants to add new kind of public transport.

Precondition: Technical staff should be logged in.

Postconditions: New kind of public transport is added to system.

Main Success Scenario:

1. Technical staff clicks onto the button which says “add new type of transport”.
2. Technical staff fills in the form with the required information about new kind of transport.
3. The system displays a text which says "Adding is successful".

Extensions:

2.a) If there are some necessary fields which were left empty,

· System will warn the technical staff that they need to fill those fields out.

· Technical staff refills the missing information and sends this information to the system.

2.b) System previously added this kind of transport.

· The system will send alert to the technical staff.

3.a)The system consists of a technical error.

· The system will send alerts to the technical staff.

Use Case UC6: Add public transport instances

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Technical staff

Stakeholders and Interests:

Technical staff: Technical staff wants to add public transport instances.

Precondition: Technical staff should be logged in.

Postconditions: Public transport instance is added to system.

Main Success Scenario:

1. Technical staff clicks into “add public transport instances” button.
2. Technical staff fills the required information about the public transport instance.
3. The system displays a text which says "Adding is successful".

Extensions:

2.a) If there are some necessary fields which were left empty,

· System will warn the technical staff that they need to fill those fields out.

· Technical staff refills the missing information and sends this information to the system.

2.b) System previously didn’t add this kind of transport.

· The system will send alert to the technical staff.

2.c) System previously added this public transport instance.

· The system will send alert to the technical staff.

3.a)The system consists of a technical error.

· The system will send alerts to the technical staff.

Use Case UC7: Add local special rules for public transports

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Technical staff

Stakeholders and Interests:

Technical staff: Technical staff wants to add local special rules for public transports.

Precondition: Technical staff should be logged in.

Postconditions: Local special rules are added to the system.

Main Success Scenario:

1. Technical staff clicks the “add special rules for public transports” button.
2. Technical staff fills the required information about the local special rules for public transports.
3. The system displays which says " Special rules has been successfully added ".

Extensions:

2.a) If there are some necessary fields which were left empty,

· System will warn the technical staff that they need to fill those fields out.

· Technical staff refills the missing information and sends this information to the system.

2.a) System previously added this local special rule for public transport.

· The system warns the technical staff by saying this special rule already exists.

Use Case UC8: Remove local special rules for public transports

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Technical staff

Stakeholders and Interests:

Technical staff: Technical staff wants to remove local special rules for public transports.

Precondition: Technical staff should be logged in.

Postconditions: Local special rules are removed from the system.

Main Success Scenario:

1. Technical staff clicks the section where it says “remove special rules for public transports”.
2. The list of special rules will be displayed on the screen.
3. The technical staff will select the special rule that he/she wants to delete.
4. The system will display a text which says "The special rules has been successfully removed".

Extensions:

2). If there is no special rule in the system,

· The system shows a warning message which basically says “‘there’s no special rule to be deleted.’

Use Case UC9: Update local special rules for public transports

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Technical staff

Stakeholders and Interests:

Technical staff: Technical staff wants to update local special rules for public transports.

Precondition: Technical staff can be login.

Postconditions: Local special rule is updated by the system.

Main Success Scenario:

1. Technical staff clicks the section where it says “update special rules for public transports”.
2. The technical staff will select the special rule that he wants to update.
3. Technical staff will modify the required information about the local special rules for public transports.
4. The system will display "Updating local special rules is successful".

Extensions:

2). If there’s no special rule in the system,

· System will warn the technical staff by saying that there’s no special rule to update

3) If there are some necessary fields which were left empty,

· System will warn the technical staff that they need to fill those fields out.

· Technical staff refills the missing information and sends this information to the system.

Use Case UC10: Synchronizing program information between devices

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Passenger

Stakeholders and Interests:

Passenger: Passenger wants their devices to be synchronized.

Precondition:

· The user should have a registered account.

Postconditions:

· Favorite paths selected by the user should be displayed quickly.
· The user should give feedback and see his / her name on the feedback list.

Main Success Scenario:

1. Guest views the login screen.
2. The system sends a form for logging in.
3. Guest fills in this form.
4. Guest presses the Login button.
5. Guest is logged into the system.
6. Guest will press the synch. button.
7. Devices will be synchronized.

Extensions:

4.a) If members cannot be found

· The system will warn the guest.

· Guest are redirected to the login screen.

6.) If there is no internet connection, system will warn the user that there is no internet connection and synchronization will fail.

For all steps, the system is unavailable due to some technical issue,

· The system will let guest know that system is down for the time being.

Use Case UC11: Giving feedback

Scope: Public Transport Route Finder System (PTRFS)

Level: User-Goal

Primarily Actor: Passenger

Stakeholders and Interests:

Passenger: Passenger wants to give feedback regarding the system.

Precondition:

- User should have an internet connection.

Postconditions:

- The system will receive the feedback.

Main Success Scenario:

1. Passenger will open the application.
2. Passenger will press the feedback button.
3. Passenger will fill in the form for the feedback.
4. Passenger will press the “send the feedback” button.
5. The system will display a text which says “the feedback has been sent successfully.”

Extensions:

3) If there are some necessary fields which were left empty,

· System will warn the passenger that they need to fill those fields out.

· Passenger refills the missing information and sends this information to the system.

For all steps, the system is unavailable due to some technical issue,

· The system will let guest know that system is down for the time being.

9) Domain Model as a UML Class Diagram