RAILWAY INFORMATION SYSTEM

CENG 491
Computer Engineering Design

Initial Design Report

BUGBUSTERS

18.12.2009

Table of Contents

1. Introduction

Railway Information System is intended to simplify the current complicated system used at Turkish State Railways. Railway Information System will not only ease the work for system administrators and machinists; but also it will improve and add new features to the online system for passengers.

a. Purpose of the Document

In this document Railway Information System structure will be explained. This structure will be constructed such that it will satisfy the requirements mentioned in Software Requirements Specification.

In Software Requirements Specification, Railway Information System's general features and requirements are mentioned. In this document, they will be analyzed in detail.

b. Scope of the Document

This document covers the architectural design, data design, procedural design, design constraints, development schedule. Also hardware and software requirements and working environment will be explained.

c. Abbreviations

RIS   : Railway Information System
IMA   : Intelligent Machinist Assignment
OTSS   : Optimum Train Suggestion System

Some words in railway jargon:
Machine   : The locomotive.
Brevet   : The certificate to use a machine
Livre   : The book containing speed/position tables

2. Project Description

a. Project Title

The title of the project is "Railway Information System".

b. Detailed Problem Definition

The system's (currently being used at TCDD) major problem is that there are many daily jobs and most of them are done manually. Secondly some features of this system are not public, which might be used to increase the public attention to railways if they were. Another problem is that some services already in use in abroad have not been introduced to TCDD customers yet.

i. There are many kinds of machines and a machinist is required to have the special brevet to use a machine. Everyday hundreds of machines goes somewhere to somewhere and every machine must be controlled by a machinist having the appropriate brevet. In current system, the assignment of machine – machinist is done daily and manually. Machinists learn the next day's schedule just 24 hours ago.

ii. All railways in Turkey are partitioned and for every partition the speed limit are determined. All these information are listed on tables in a book called *Livre*. When a machinist starts the journey from some point, he/she (he will be used later for convenience) opens the relevant page of Livre. Near the railways, there are position signs and with the help of them, machinists know their positions. Knowing the position, a machinist looks up the speed limit of that position from the Livre and adjusts the speed accordingly. Machinists have to follow both the position signs and the speed limit tables from beginning to the end of the journey.

iii. Current system has been developed as time passed and new technologies appeared. So it has been always built on the old one. Each time taking care to preserve the backward compatibility, adding new features have caused the current system to be very complicated and dirty.

iv. About 15 trains of TCDD already have a GPS device. But the information fetched from these are visible to only some supervisors, even machinists cannot see them.

v. In abroad, there exists some optimum cost / optimum time train suggestion systems [1] but there is not a similar system in Turkey.

c. Project Definition and Goals

Railway Information System is a project to enhance the service quality of TCDD for machinists and administrators at TCDD and passengers by eliminating the problems stated above.

The aims of the RIS project include:

- Design a better train control system
  - o Automate the machinist – machine matching (IMA)
  - o Remove the "checking the speed limit from Livre" and replace it with "checking needed information from machinist panel"
  - o In case of an emergency, establish a faster and easier communication with control center and deliver the train position more accurately

- Attract public interest on railways
  - Make the train position tracking public
  - Implement a optimum time / optimum cost train suggestion system
  - Meeting this goal means highly use of public transportation, less traffic jam and accidents

- The use of system will be made easier for all three categories of users (passengers, machinists, administrators) with knowledge database, help pages and manuals.

3. Design Constraints and Working Environment

a. Design Constraints

i. Time

In order to meet the deadlines and complete the project on time, team members have to strictly obey the time constraints stated on Gantt chart.

ii. Versioning

As soon as the implementation starts, different states of process have to be versioned. SVN will be used as the version control system.

iii. Financial Constraints

All implementation software and tools are open source except AutoCAD Map 3D. AutoDesk gives free license for this program to students.

b. Working Environment

The system will be implemented on *Java* language. Users will be able to reach the system within a browser page, so a java running server will be needed. *Apache server* [12] and *Apache Tomcat* [7] will be used for that purpose. The map implementation will be done on *AutoCAD Map 3D* [13]. For GPS simulation, java libraries will be used. *NetBeans IDE*[4] will be used as the development environment.

4. Use Case Analysis

a. Registering New Users
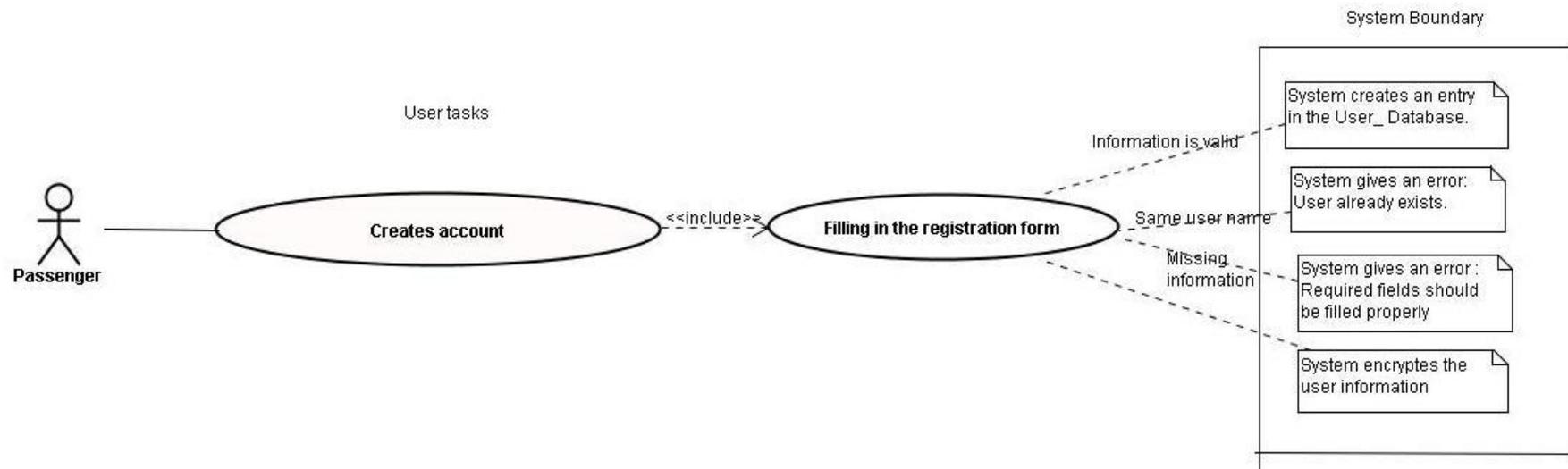
- A new passenger who has not an account yet clicks to the link "Yeni Kullanıcı" on the Login Page.

- The registration page will be opened. The passenger fills the registration form and click "Tamam" button to create an account

- Passengers can cancel this process by clicking "İptal" button or they can clear all fields by clicking "Temizle" button

- System creates a new entry in the passenger database if the required areas are filled properly.

- Otherwise:

  o The system shows an error message if there is an empty required field in the registration form.
  o The system shows an error message if the username the passenger types in registration form already exists.

- Administrators create their own accounts and machinists' accounts directly by executing a query on the database table.

- System encrypts all user information properly.

b. Logging into the System
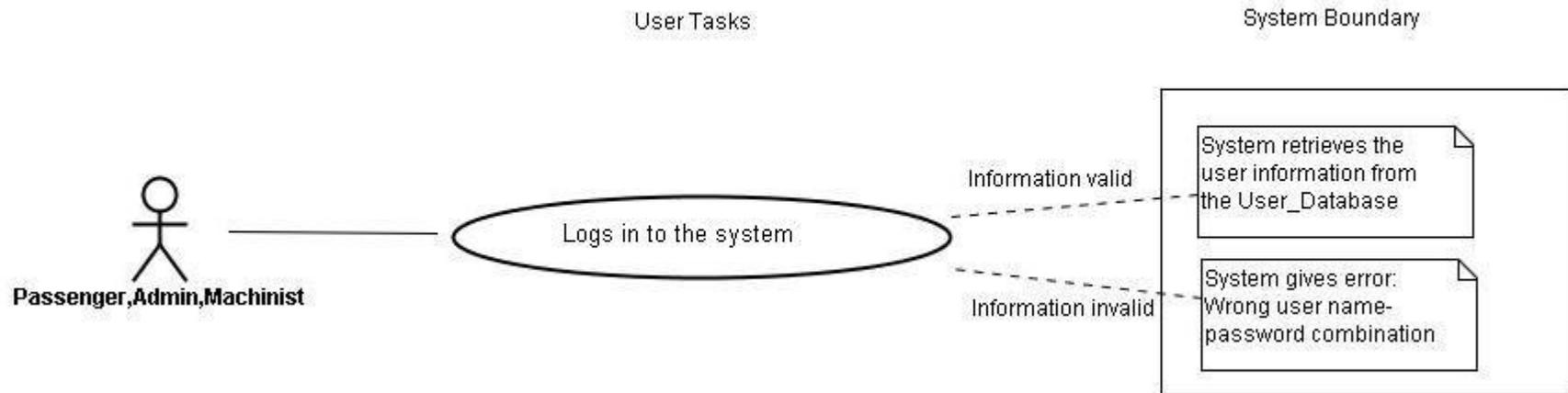
- Users logs into the system via Login Page web interface. To log into the system, a user provides his username, password and user type.

- There are three user types: admin, machinist and passenger.

- After filling username and password areas and selecting the user type, a user clicks Login button for confirmation and Cancel button for cancellation.

- System will give an error if it cannot find such username, password, user type combination. Otherwise system lets the user log in and access user type related information.

c. Updating Personal Information
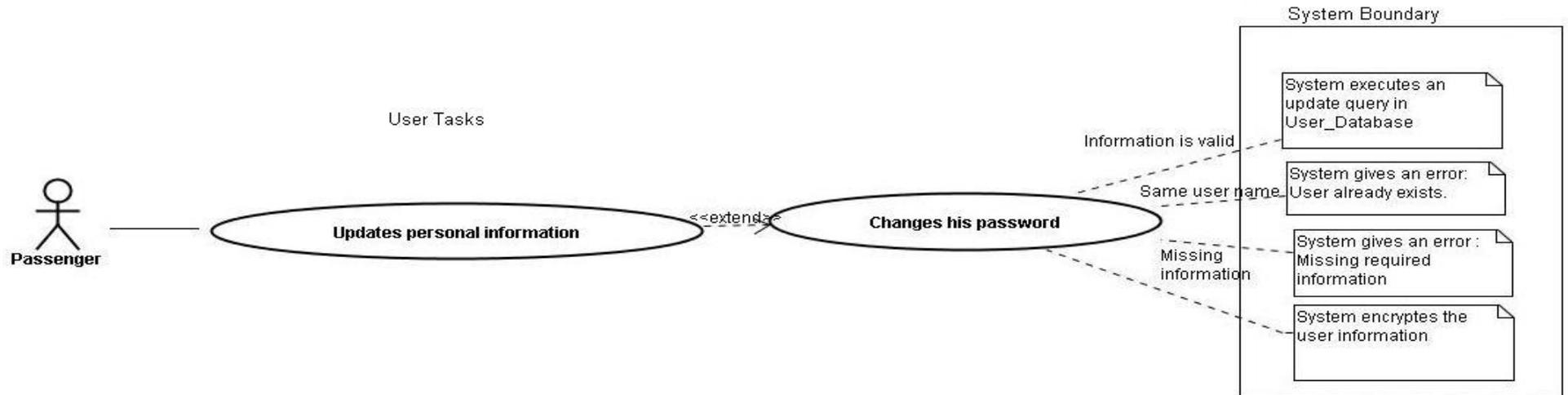
- Passenger clicks the personal information link on their home page to update their personal information.

- The "Kişisel Bilgiler" page is opened allowing the passenger to change their personal information.

- Passenger clicks on "Kaydet" button to save the changes.

- Passenger can cancel this process by clicking "İptal" button.

- The system executes an update query if all information provided meets the requirements.

- Otherwise:
    - The system shows an error message if the username the new user chooses already exists in the system.
    - The system shows an error message if there is any empty required field in the registration form.

- System encrypts all users' information properly again.

d. Live Train Monitoring Page for Passengers
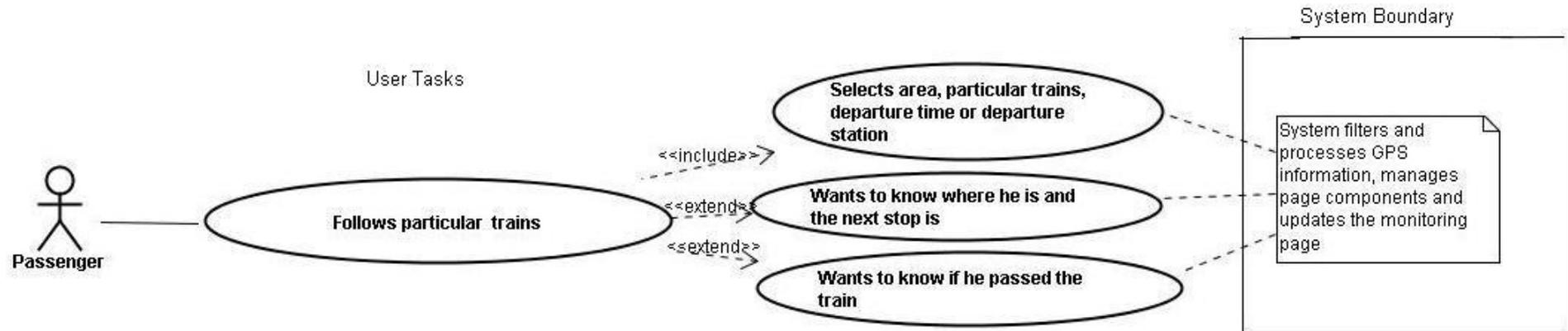
- Passenger clicks the "Tren Takip Sistemi" link on his home page to follow a train.

- The "Tren Takip Sistemi" page is opened allowing users to decide a specific railway and train.

- Passenger clicks the "Seçtiğiniz Hat" combo box to choose a specific railway. System updates the other combo boxes, text fields and the map according to this decision. The "Seçtiğiniz Tren" and the "Çıkış Yeri" combo boxes become enabled. Text fields remain empty.

- Passenger clicks the "Seçtiğiniz Tren" combo box to choose a specific train on this railway. System updates the other combo boxes, text fields and the map according to this decision. The "Çıkış yeri " and "Çıkış Saati" combo boxes become disabled and are set by the system automatically. All text fields are set automatically too and the map focuses on this specific train. After this point simulation begins and this page is updated periodically according to the given information from the GPS of this train.

- Passenger clicks the "Çıkış Yeri" combo box to choose a specific on this railway. "Seçtiğiniz Tren" combo box becomes disabled and "Çıkış Saati" combo boxes remains enabled. System doesn't set any text field until the user determines the departure time of the train.

- Passenger clicks the "Çıkış Saati" combo box to complete the needed information to specify a train. After the user specifies departure time of the train, all text fields and the "Seçtiğiniz Tren" box are set automatically and the map focuses on this specific train. After this point simulation begins and this page is updated periodically according to the given information from the GPS of this train.

e. **Live Train Monitoring Page for Machinists**
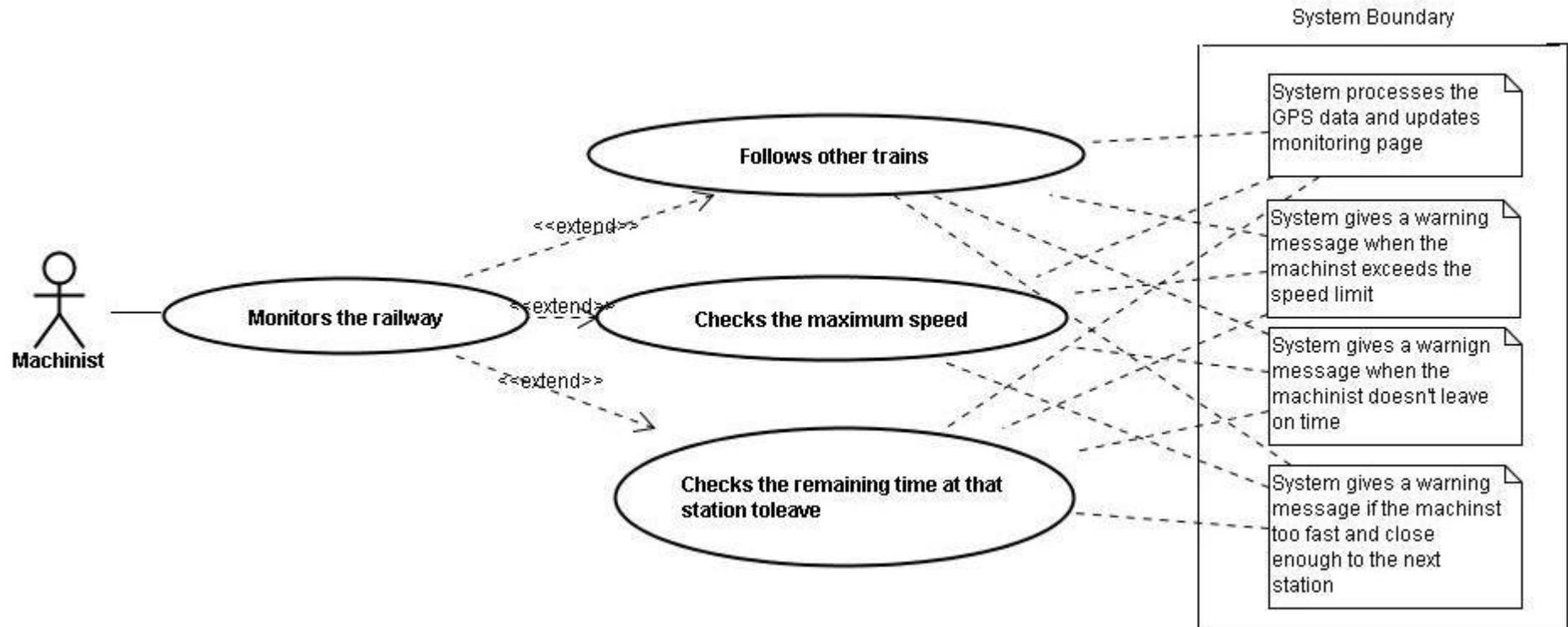
- Machinist clicks the monitoring link "Tren Takip Sistemi" on his home page to access the monitoring page.

- The "Tren Takip Sistemi" page is opened allowing machinist to watch the train and get current information.

- System gives a warning message if the machinist exceeds the speed limit.

- System gives a warning message if the machinist stays longer than expected at a station.

- System gives a warning message if the train is still very fast but not far away from the next station.

f. **Travel Planning System Use Case**
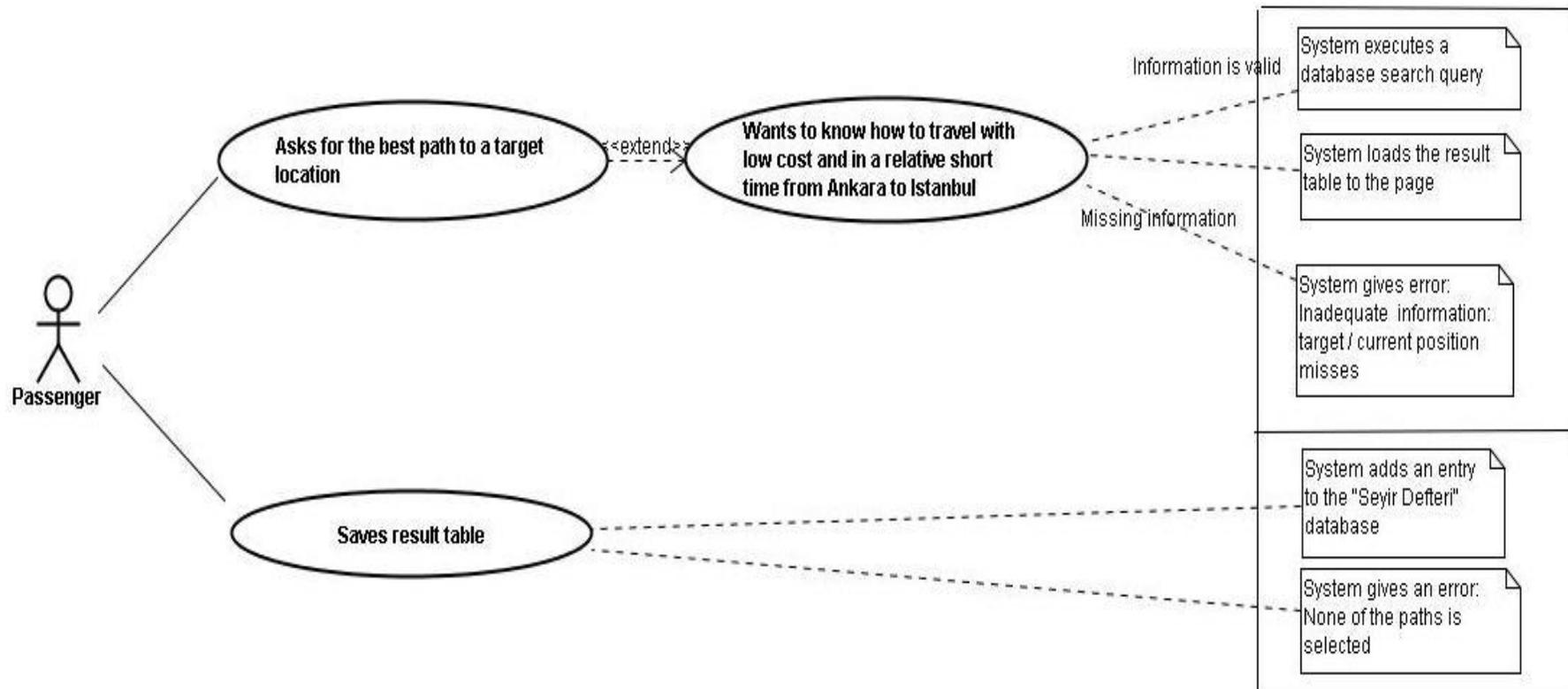
- After passenger logs into the system, "Ana Sayfa" is opened.

- Passenger fills in the form according to his travel plan and presses the "Listele" Button.

- System executes an optimized search query on the database table "connects" to find the appropriate travels for this specific destination and target if there isn't any missing required field like target or destination.

- Otherwise, system gives error message "Inadequate information".

- Passenger clears all of the fields by pressing "Temizle" button.

- After a successful query operation, system loads the table of results to the end of the page and 2 more buttons "Kaydet" and "Sil" appear.

- Passenger chooses some of the results from the table and clicks "Kaydet" to save the chosen results.

- System gives an error message if none of the results is chosen.

- Otherwise system adds an entry to the "Best_Path" database.

- Passenger clicks the "Sil" button to clear this table from this page. System clears this table form the page.

g. **News Page Use Case**
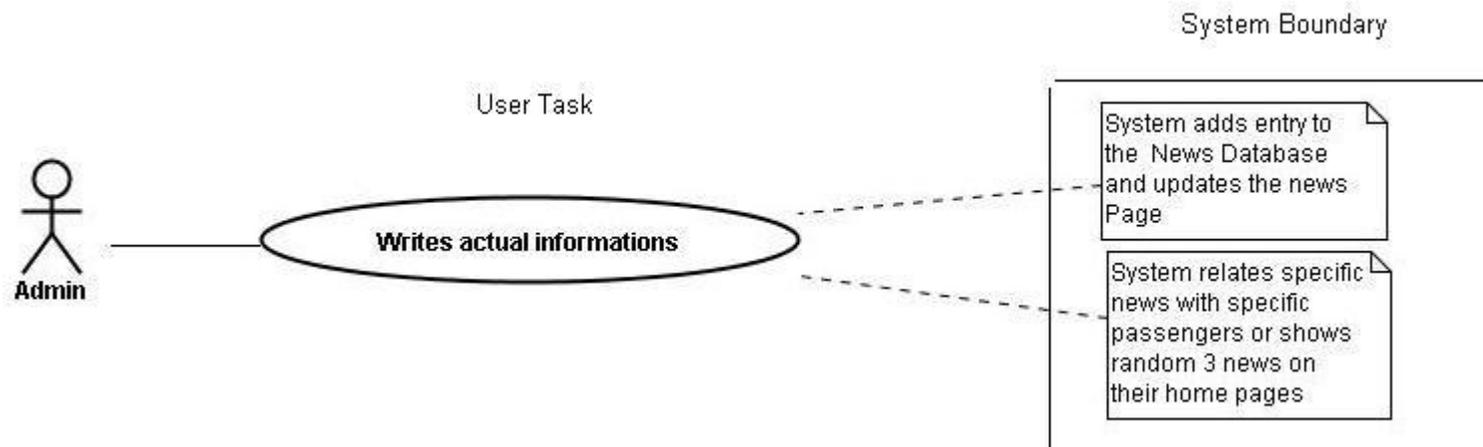
- Administrator creates a "news entry" in the system via the interface of News Database.

- System adds an entry to the table and updates the News Page, which is only available to the administrators in the system.

- System relates this news with the passengers according to their location and preferred trains.

- If no such a passenger is found, system shows 3 random news entries, which are relatively newer than the other news on the home page of passengers.

h. Knowledge Database Use Case

Passenger

User task

System Boundary

Searches how to change his password

Seaching in the FAQ list

Searching in the General Questions and Answers list

Searches solutions in the system

Searches in the Pending Questions list

Searches via search button

<<extend>>

<<include>>

<<include>>

<<include>>

<<include>>

System executes a search query in Knowledge_Database

System filters the results according to the preference of passenger (solved,pending..)

System shows the results in a seperate page.

System gives error: No matching question

- Passenger clicks "Sorun Giderme" button on the home page and the "Bilgi Sistemi" Page is opened.

- Passenger clicks a related topic in the section "Sıkça Sorulan Sorular", which are given as links in this page, to find a solution to his question.

- Passenger finds a similar question and clicks to it to see the details.

- The chosen problem is displayed in a problem specific page.

- If there isn't any related topic to his question, passenger returns to the "Bilgi Sistemi" Page and clicks "Diğerleri" link at the end of this category.

- The page of problems, which don't belong to any general topic is opened.

- Passenger clicks to the similar problem if such a problem exists.

- If there isn't any relevant question in this section, user returns to the "Bilgi Sistemi" Page and writes some keywords related to his question into the text field reserved for search purpose and clicks "Ara" button.

- The results are listed in a separate page if there is any.

- System gives an information message "No matching question " if there isn't any question related to these keywords.

- This operation can also be repeated with other categories: "Genel Sorular ve Çözümler" or "Çözüm Bekleyen Sorular".

i. Creating a New Question Use Case
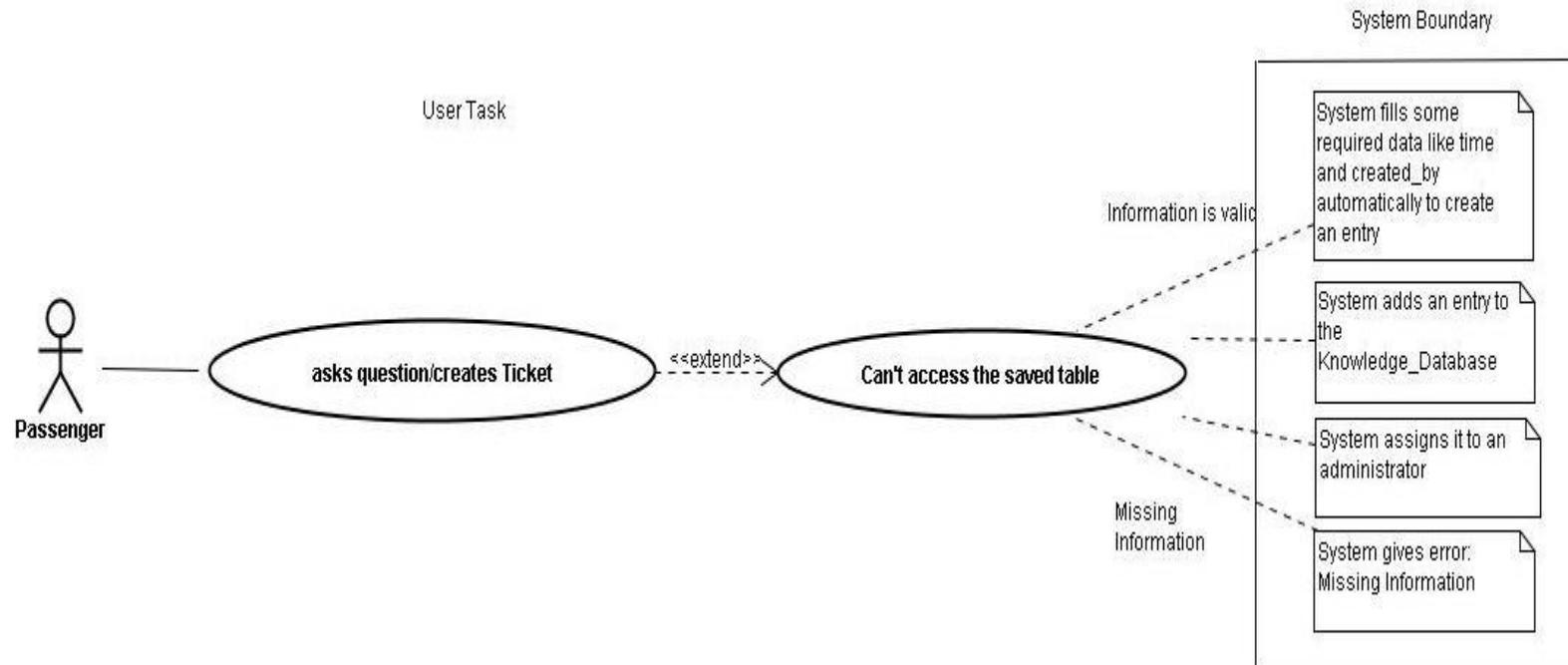
- If the search operations result in failure, passenger clicks the button "Yeni Soru".

- The page of "Yeni Soru Girişi" is opened allowing passenger to write his question and some related information

- Passenger fills the form of new question and clicks "Gönder" button to send his question to the administrators.

- System creates an entry in the Knowledge Database if the required fields like problem title and description are filled. Otherwise system gives error "Missing Information".

- System assigns this problem to an administrator whose job description includes problem solving.

- Passenger clicks "İptal" button to cancel this process at any time.

- Passenger clicks "Temizle" button to clear all fields in the ticket creation page.

j. **Schedule Page Use Case**
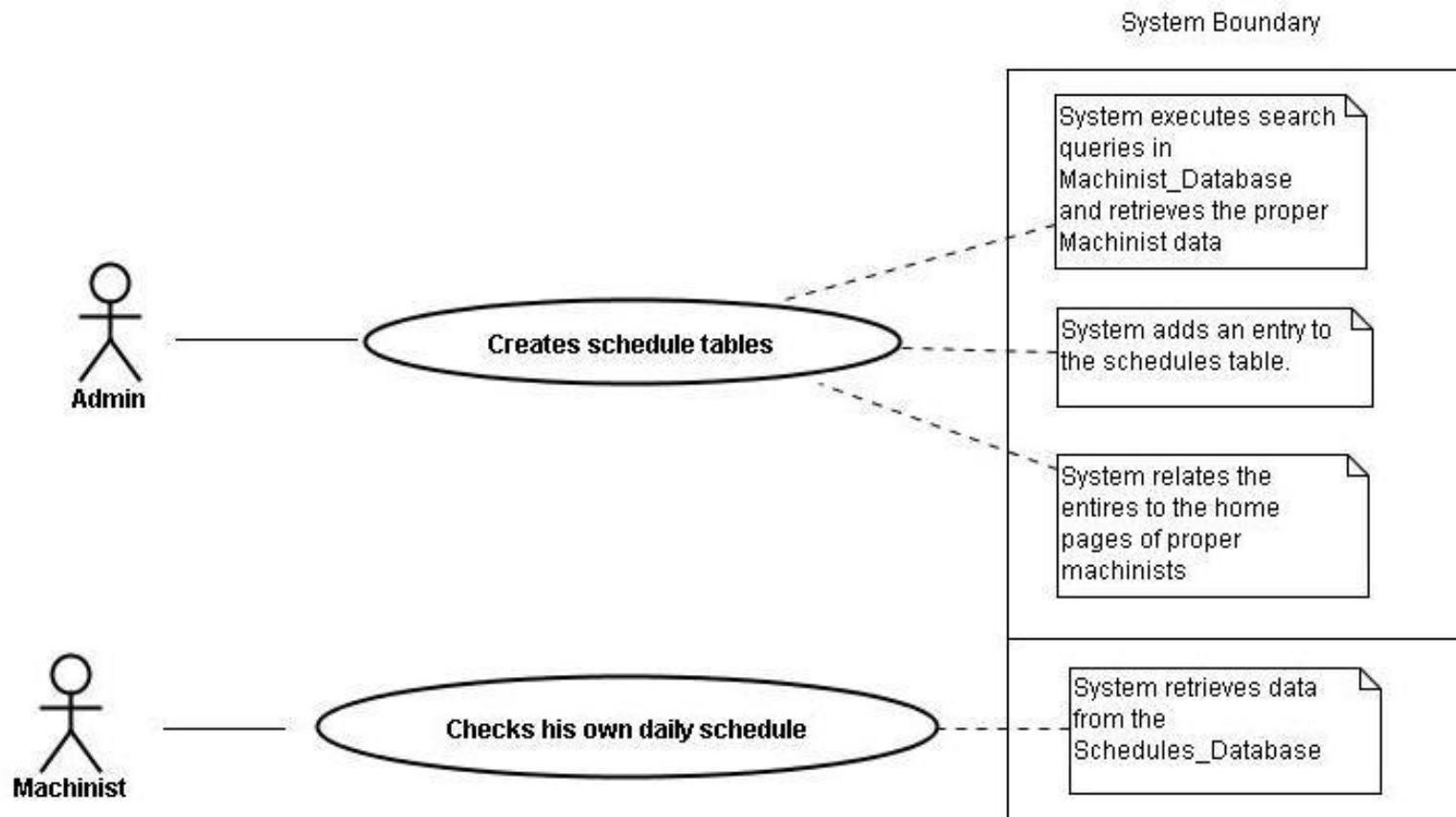
- Administrator executes a search query to find proper Machinists who can drive a specific train in a specific time interval.

- Administrator creates schedule tables via database interfaces.

- System creates a schedule table which stores daily schedules of all machinists.

- Machinists logs into home page to see his daily train schedule.

- System relates these entries in the schedule table with the proper machinists and loads them to the home page of machinist.

5. Data Design
a. Entity – Relationship Diagram
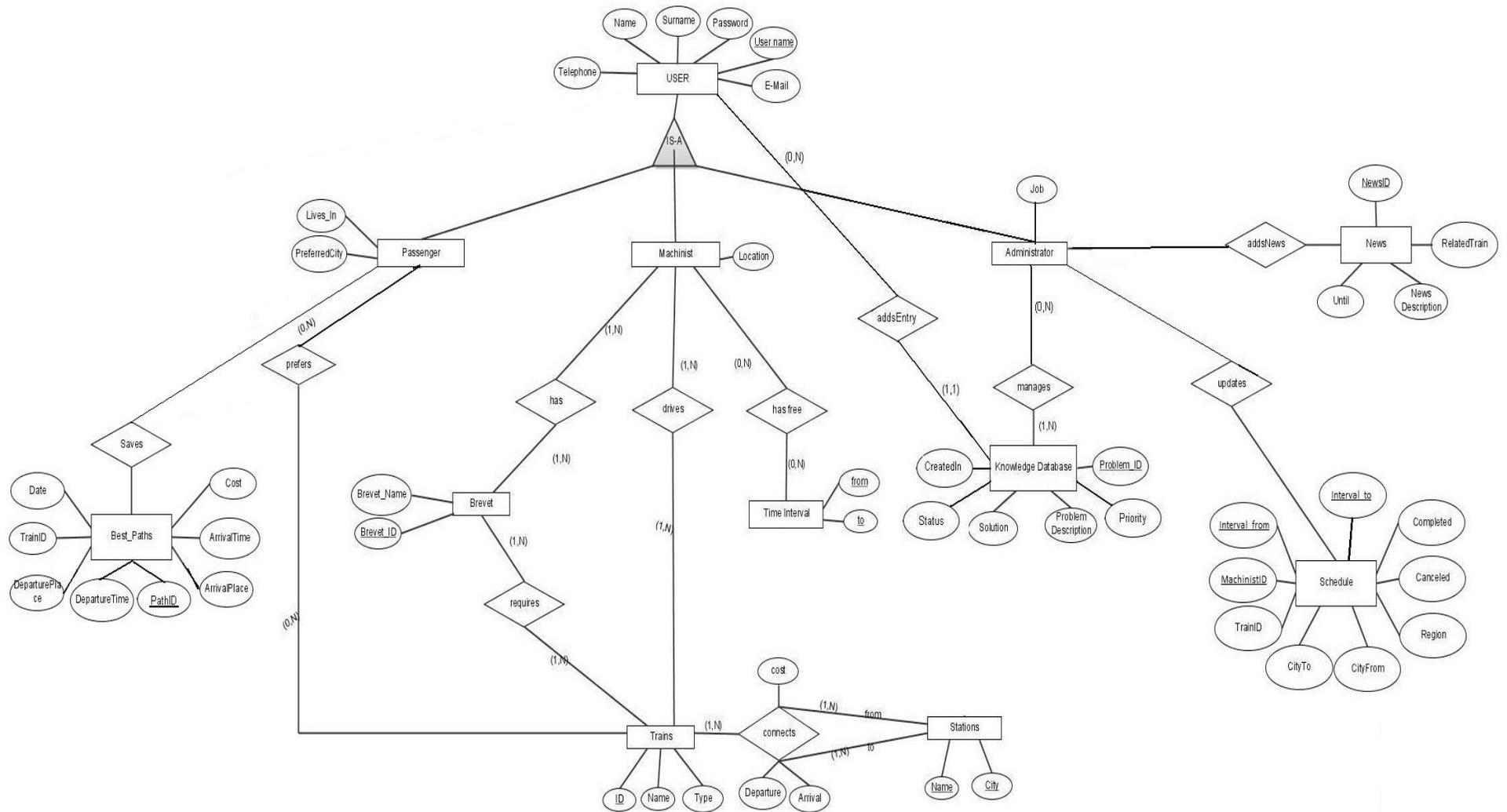
b. Database Schemas
i. USER table

Field	Type	Null	Foreign Key	References
User_name (p.k.)	Varchar(20)	no	no	-
Name	Varchar(20)	no	no	-
Surname	Varchar(20)	no	no	-
Password	Varchar(20)	no	no	-
E-mail	Varchar(30)	no	no	-
Telephone	Integer	no	no	-

This table holds user information. Every user has to supply name, surname, password, e-mail, and telephone information and select a username. Following three tables extends from this table.

ii. Passenger table

Field	Type	Null	Foreign Key	References
LivesIn	Varchar(20)	no	no	-
PreferredCity	Varchar(20)	no	no	-
User_name (p.k.)	Varchar(20)	No	Yes	USER

This table extended from USER table and every passenger has to supply the city he lives in and the city where he wants to prefer to go.

iii. Machinist table

Field	Type	Null	Foreign Key	References
Location	Varchar(20)	no	no	-
User_name (p.k.)	Varchar(20)	No	Yes	USER

This is another table extended from USER table and every machinist's location information is kept on that table.

iv. Administrator table

Field	Type	Null	Foreign Key	References
Job	Varchar(20)	no	no	-
User_name (p.k.)	Varchar(20)	No	Yes	USER

This is the third table extended from USER table and every administrator's job is kept on that table. A job may be adding solutions to Knowledge_Database, adding entries to News database, editing Schedule database, or just answering phones without any write access to any database.

v. Best_Paths table

Field	Type	Null	Foreign Key	References
PathID (p.k.)	Integer	no	no	-
Date	Date	no	no	-
TrainID	Integer	no	yes	Trains
DeparturePlace	Varchar(20)	no	no	-
DepartureTime	Time	no	no	-
ArrivalPlace	Varchar(20)	no	no	-
ArrivalTime	Time	no	no	-
Cost	Float	no	no	-
UserID	Varchar(20)	no	yes	USER

This table contains information about a passenger's best path searches. After a user searches some path, it is kept on that database and users may retrieve their past searches.

vi. Brevet table

Field	Type	Null	Foreign Key	References
Brevet_ID (p.k.)	Integer	no	no	-
Brevet_Name	Varchar(20)	no	no	-
AllowedTrainTypes	Varchar(200)	no	no	-

Machinists are required to have the appropriate brevet to use a machine. In that table, besides the brevet ids and brevet names matching, the allowed train types for every brevet type are already kept in that table.

vii. Trains table

Field	Type	Null	Foreign Key	References
ID (p.k.)	Integer	no	no	-
Name	Varchar(20)	no	no	-
Type	Varchar(20)	no	no	-

All the trains are kept in that table. Also the train's machine type is kept. It will be used while assigning a machinist to use that train.

viii. Stations table

Field	Type	Null	Foreign Key	References
Name (p.k.)	Varchar(20)	no	no	-
City	Varchar(20)	no	no	-

On that table, every station is given a unique name and the city information is kept.

ix. Time_Interval table

Field	Type	Null	Foreign Key	References
from (p.k.)	Timestamp	no	no	-
to (p.k.)	Timestamp	no	no	-
MachinistID	Varchar(20)	no	yes	USER

On that table, every machinist's free time intervals are kept.

x. Knowledge_DB table

Field	Type	Null	Foreign Key	References
ProblemId (p.k.)	Integer	no	no	-
Problem_Description	Varchar(1000)	no	no	-
Status	Integer	no	no	-
Priority	Integer	no	no	-
Solution	Varchar(2000)	yes	no	-
CreatedIn	Timestamp	no	no	-

That table keeps the problems reported and the solutions to that problems as soon as administrators edits the entries and add solutions. Status field indicates whether the problem waits for an answer or it is solved. High priority problems will be handled firstly.

xi. News table

Field	Type	Null	Foreign Key	References
NewsID (p.k.)	Integer	no	no	-
RelatedTrains	Integer array	yes	yes	Trains
News_Description	Varchar(2000)	no	no	-
Until	Timestamp	yes	no	-

This table holds data about some special offers such as a cut on ticket prices. They will be showed to passengers as advertisements. Also in that table, the delay or cancellation information will be hold.

xii. Schedule table

Field	Type	Null	Foreign Key	References
Interval_from (p.k.)	Timestamp	no	no	-
Interval_to (p.k.)	Timestamp	no	no	-
MachinistID (p.k.)	Varchar(20)	no	yes	USER
TrainID	Integer	no	yes	Trains
CityTo	Varchar(20)	no	no	-
CityFrom	Varchar(20)	no	no	-
Region	Varchar(20)	no	no	-
Completed	Boolean	yes	no	-
Cancelled	Boolean	yes	no	-

This table contains scheduling information of all trains. Each entry holds train and machinist matching, departure/arrival time and cities and two completed/cancelled Boolean fields.

6. Class Diagram

Best_Paths
- Date : Date
- TrainID : Integer
- DeparturePlace : String
- ArrivalPlace : String
- DepartureTime : Time
- ArrivalTime : Time
- cost : Float

+ all get&set Methods() : void
+ Best_Paths() : void

USER
- User_name : String
- Name : String
- Surname : String
- E-mail : String
- Telephone : int
- Password : String

+ get& set Methods() : void
+ User(UserName : String) : void
+ logInOk(UserName : String, Password : String) : boolean
+ updatePersInfo(Index : Integer, newValue : String) : void
+ addProblem(ProbDescr : String) : void

News
- NewsID : Integer
- NewsDescription : String
- RelatedTrains : IntegerArray
- Until : Date

+ all get&setOperations() : void
+ getNewsRelatedTo(UserID : Integer) : IntegerArray

Admin
- Job : String

+ getJob() : String
+ setLocation(City : String) : void
+ Admin(Username : String) : void

Schedule
- Interval : Time_Interval
- User_Name : String
- TrainID : Integer
- CityFrom : String
- CityTo : String
- Region : String
- Canceled : boolean
- Completed : boolean

+ Schedule(I : Time_Interval, U : String, ID : Integer, C1 : String, C2 : String) : void
+ getScheduleOf(User_Name : String)(Results:ScheduleEntry) : void

Passenger
- LivesIn : String
- PreferredCity : String

+ getLivesIn() : String
+ getPreferredCity() : String
+ setLivesIn(City : String) : void
+ setPreferredCity(City : String) : void
+ Passenger(UserName : String) : void
+ logInOk(UserName : String, Password : String) : boolean
+ searchForOptimumPath(City1 : String, City2 : String) : ResultEntry
+ saveEntry(results : ResultEntry) : void
+ findProblemAbout(Keywords : StringArray) : IntegerArray
+ addProblem(ProblemDesc : String) : void

Machinist
- Location : String

+ getLocation() : String
+ setLocation(City : String) : void
+ Machinist(UserName : String) : void

hasFree
+ getFreeTimeOf(Name : String) : TimeArray

assignedTo
+ getAssignedProblems(Name : String) : IntegerArray
- getNonAssignedProblems() : IntegerArray

hasBrevet
+ getHavingMachinists(BrevetId : Integer) : StringArray
+ getHavingBrevets(UserName : String) : IntegerArray

Brevet
- Brevet_Name : String
- Brevet_ID : int

+ getBrevetName() : String
+ setBrevetName(Name : String) : void
+ getBrevetId() : String
+ setBrevetId(Integer : int) : void

Time_Interval
- from : Time
- to : Time

+ Time_Interval(from : Time, to : Time) : TimeArray

Knowledge_DB
- ProblemId : Integer
- Problem_Description : String
- Status : Integer
- Priority : Integer
- Solution : String
- CreatedIn : Time

+ All get&set Methods() : void
+ setSolution(ProblemId : Integer, String Solution : int, User_Type : String) : void

connects
- Cost : Float
- Departure : Time
- Arrival : Time

+ getTrainsConnecting(City1 : String, City2 : String) : IntegerArray
+ getTrainsFrom(City : String) : IntegerArray
+ getTrainsTo(City : String) : IntegerArray

requires
+ getRequiredBrevets(TrainID : Integer) : IntegerArray
+ getRequiringTrains(BrevetID : Integer) : IntegerArray

Stations
- Name : String
- City : String

+ getName() : String
+ setName(Name : String) : void
+ getCity() : String
+ setCity(City : String) : void

Trains
- ID : Integer
- Name : String
- Type : String

+ Trains(ID : Integer) : void

saves • addsNews • updates • assignedTo • hasFree • hasBrevet • drives • prefers • requires • connects • from • to • 0..* • 1..*

7. Interface Design
a. Registration Interface

On registration page, the user will be asked to supply some personal information and also he will be asked to supply a username – password combination. If the users marks the checkbox, he will receive e-mails about promotion news.

b. Login Interface

On login page, user types his username and password, and then selects the user type from the combo box. When user clicks the "Tamam" button, the supplied information is checked and the user will be given authorization if they are correct. Also on that page, not registered users may follow the "Yeni Kullanıcı" link to register to the system.

c. Update Information Interface

On that page users will be able to edit the personal information they supplied while registering to the system. Also users may change password on that page. If user clicks the "Kaydet" button, the personal information will be updated with the information on the form. If user click "İptal" button, there will no change on personal information.

d. Knowledge Database Interface
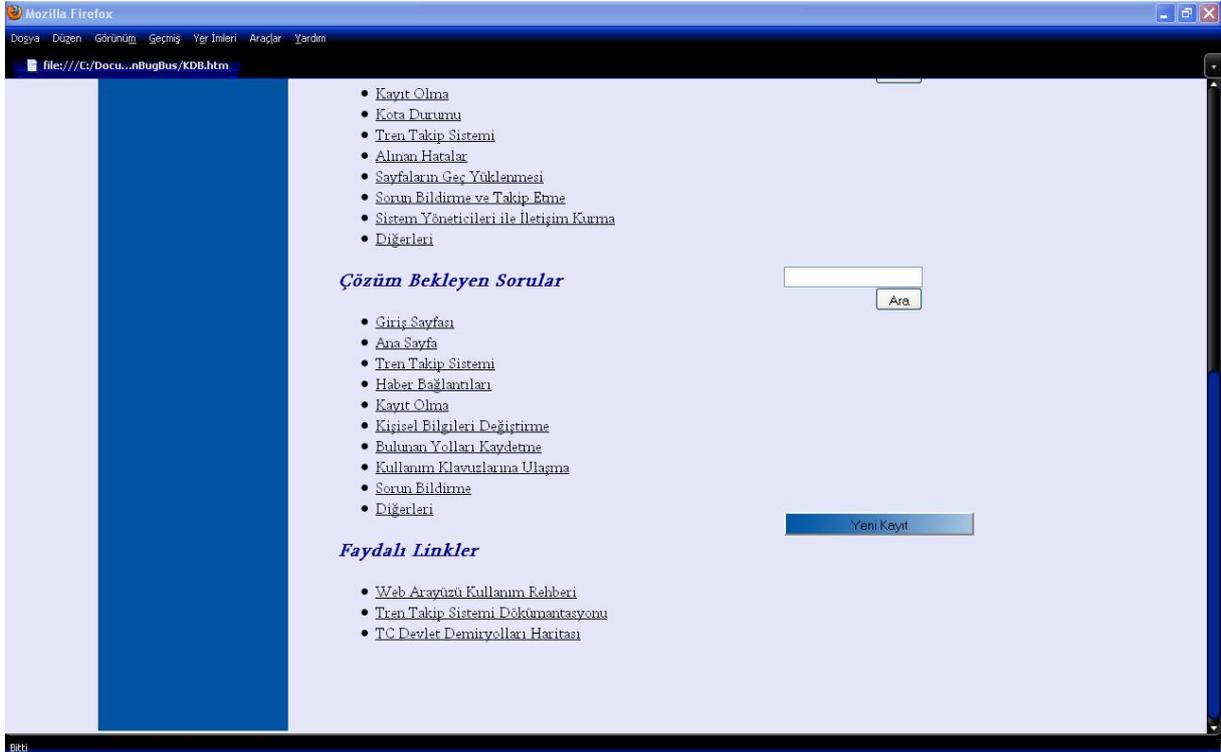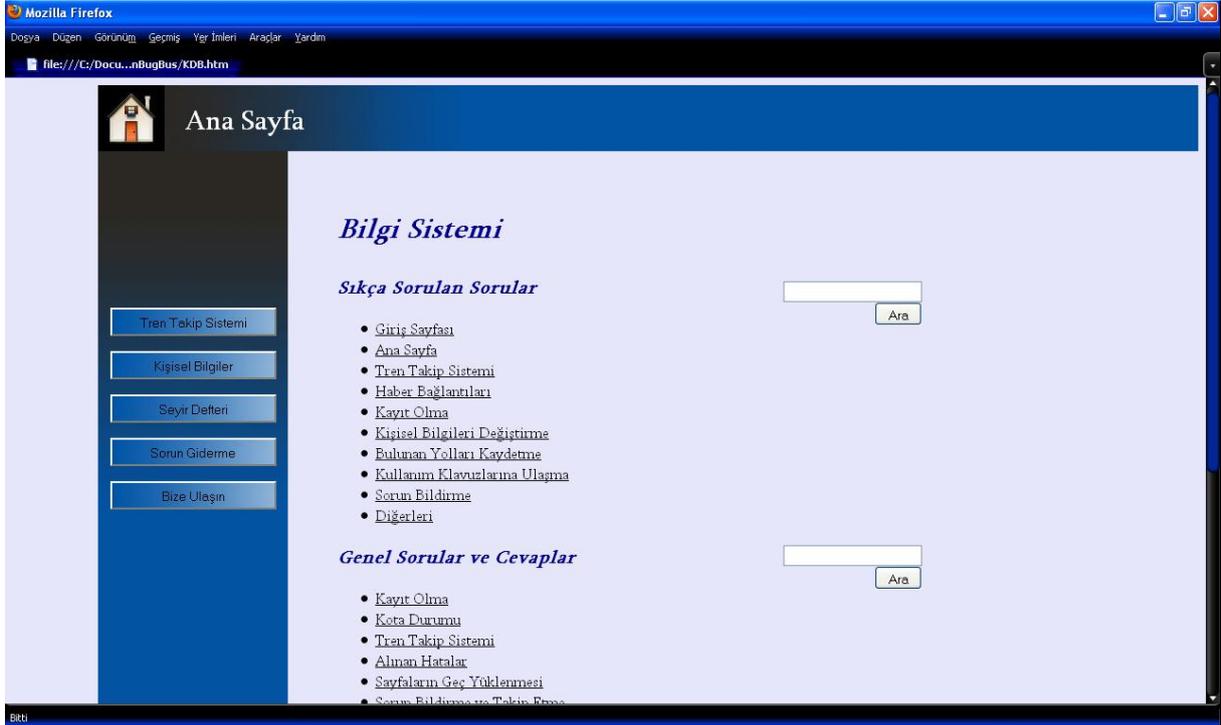
On knowledge database and help page, there are many links grouped under four sections:

- Frequently Asked Questions
- General Questions & Answers
- Questions Waiting Answers
- Useful Links

A user may use the search boxes to search some keywords on help pages. Also a user may create a new problem entry by clicking the "Yeni Kayıt" button.

When the user clicks the "Yeni Kayıt" button, a form page displayed. User is required to type a problem title and description and also select the problem category.

A user may search for an optimum cost / optimum time train. User may choose the destination place, arrival place, date and time interval, maximum price, maximum number of train changes and maximum journey time and system shows the corresponding trains.

Also the user's search histories are kept and a user may see his search histories anytime he wants.

f. Passenger Tracking Interface

On the left side of that page, a passenger may see the position of a train he selects on the map. On the right, the details about selected train are displayed.

g. Machinist Tracking Interface

A machinist may see only the positions of the trains on the same railway route. On the right half of the page, the position, next station, distance to next station, expected arrival time, expected remaining time to arrival, expected departure time, current speed, the speed limit and the distance to the next light are displayed. Also the current railway route, train id, departure city information will be on the interface. In some cases, if there is a warning, it is displayed on the middle frame of the right half of the page.

If a machinist exceeds the speed limit, an appropriate warning will be displayed and it will be written with red color to get the attention of the machinist. This frame will also be used when departure time comes and warns the machinist to move. Another use of that warning frame is while getting nearer to a station or control point, telling the machinist to slow.

h. Machinist Task Interface
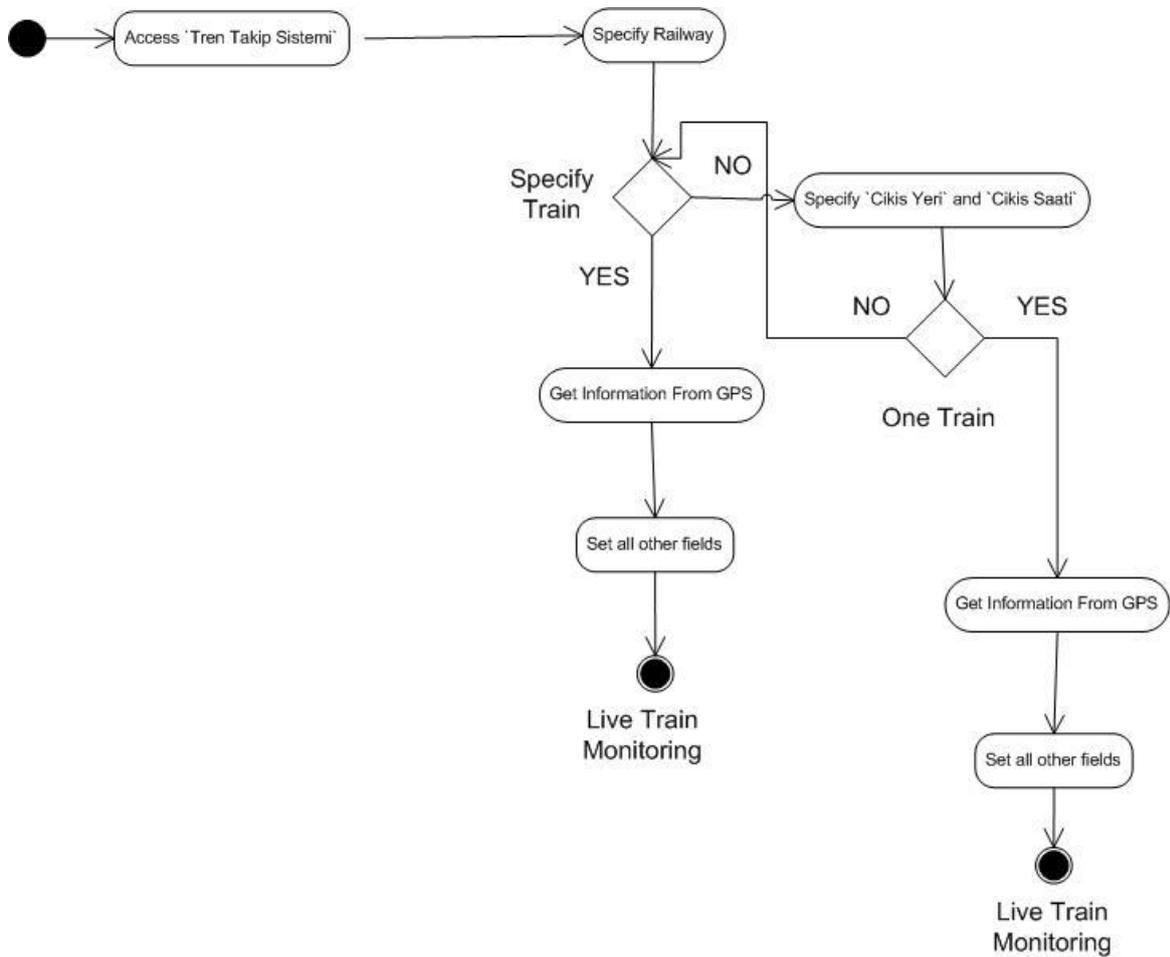

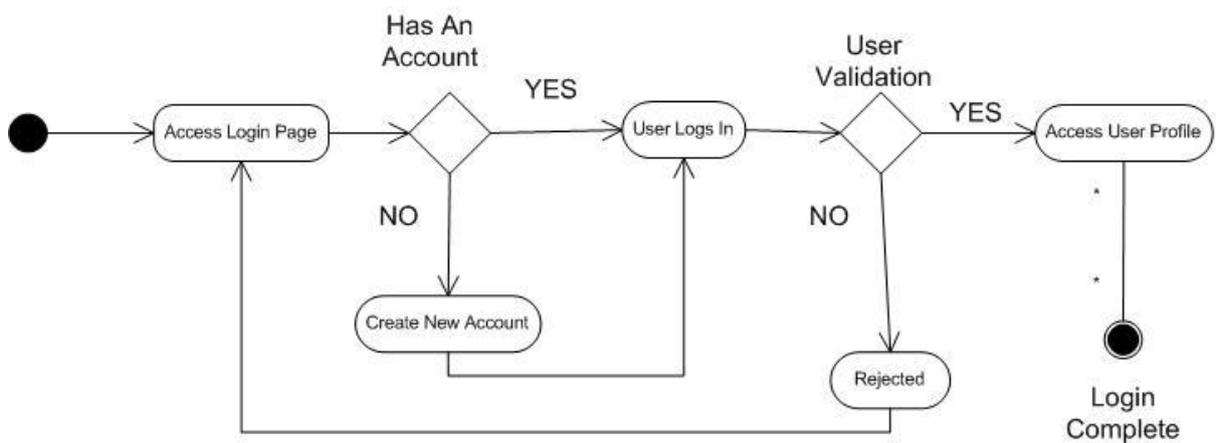Machinists can see their task schedules on that page. On the list, the train ids, regions, departure and arrival cities and times are displayed.

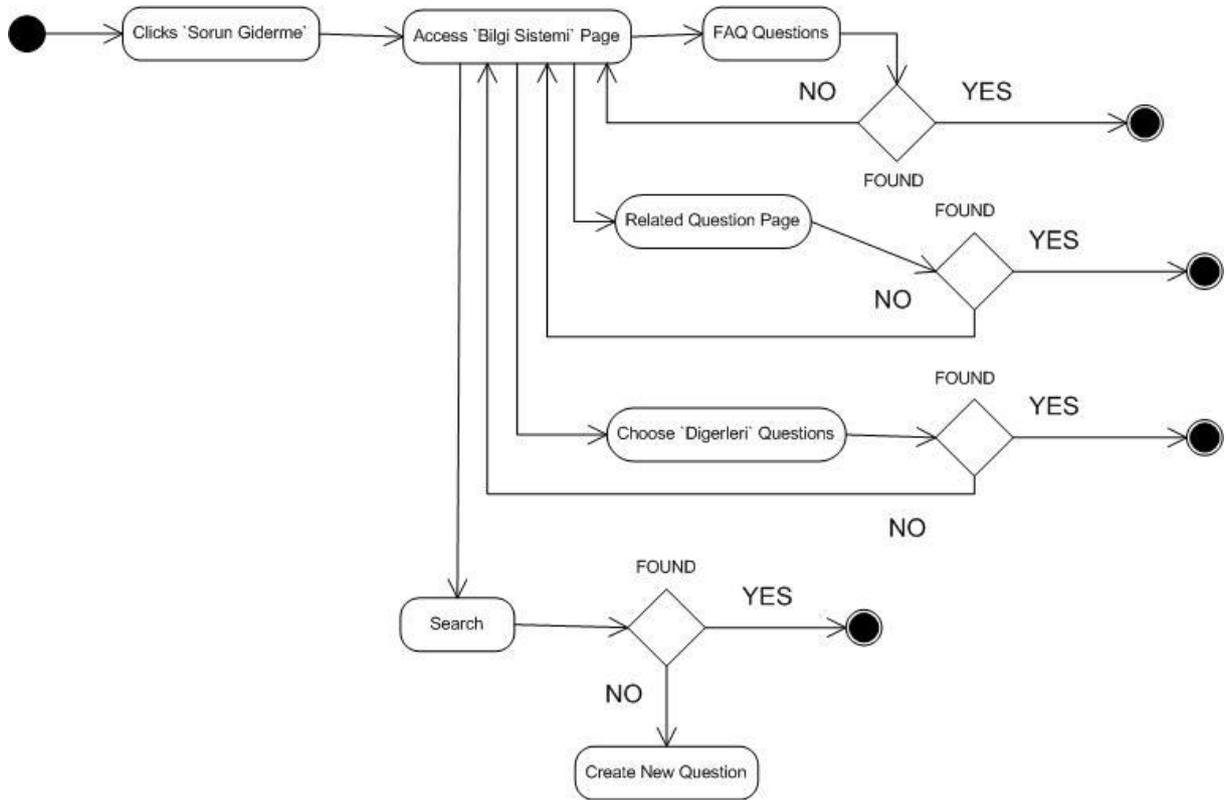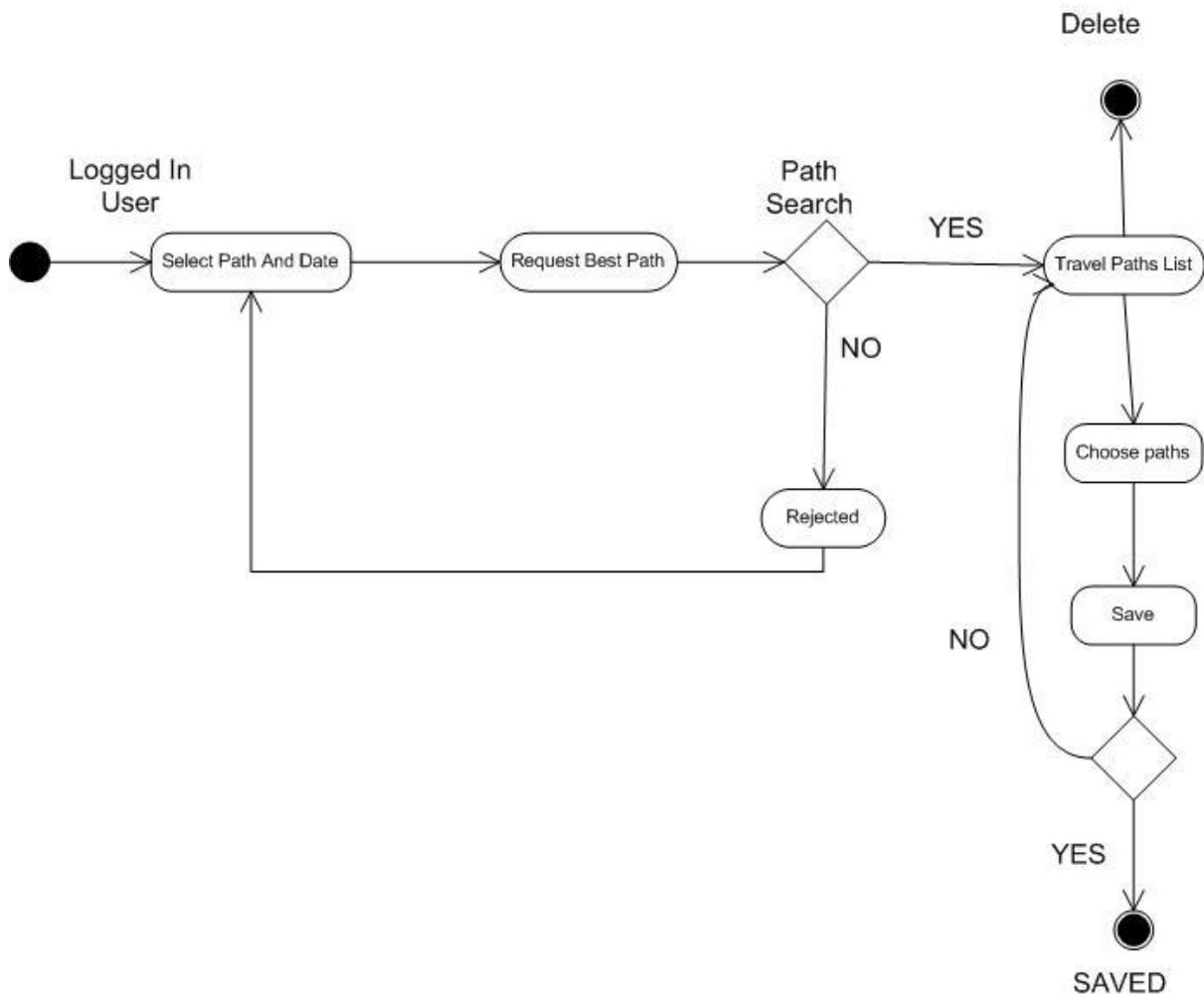## 8. State Diagrams

a. Train Monitoring State Diagram

b. Login State Diagram

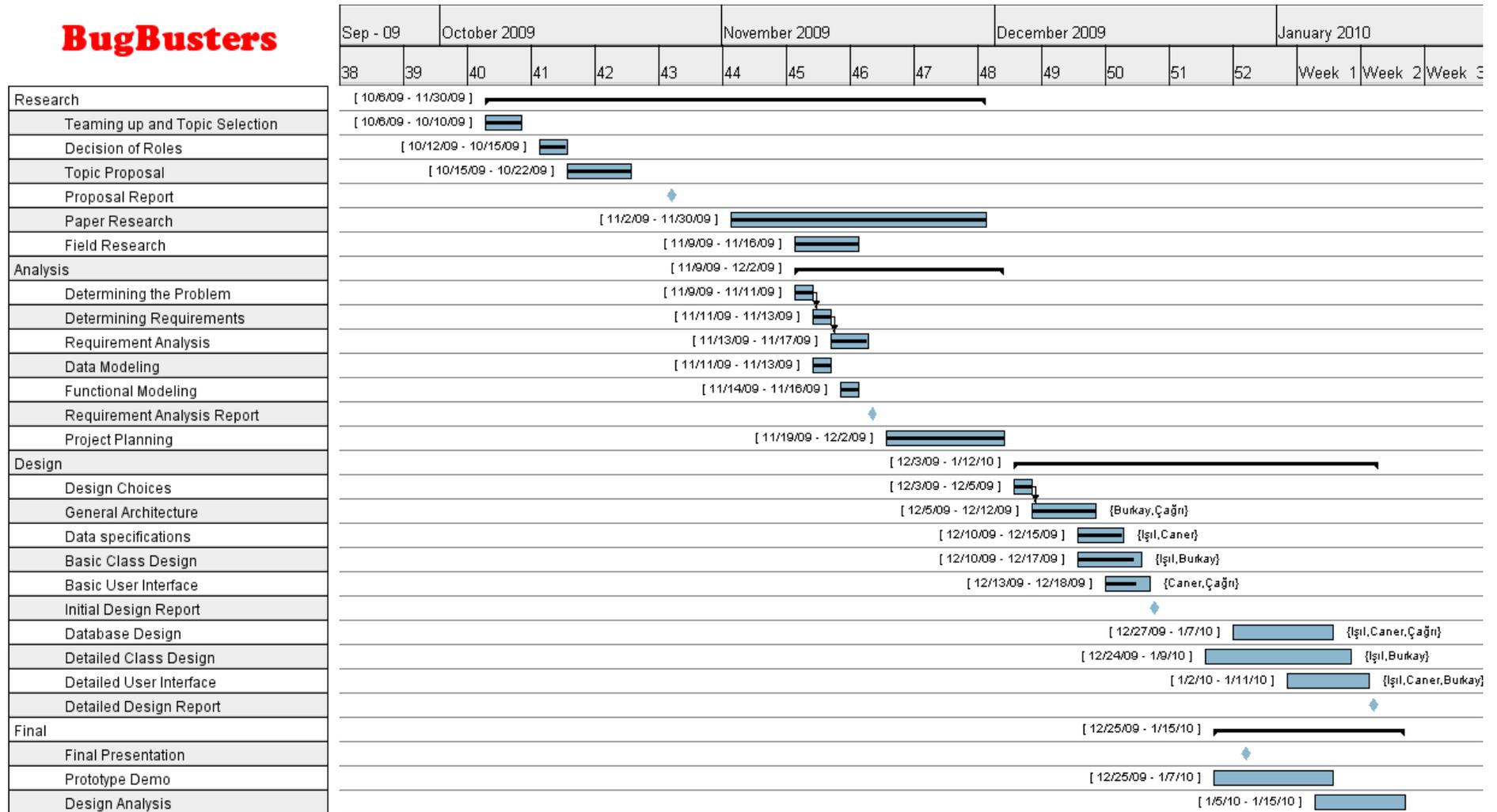### c. Adding New Question State Diagram

d. Best Path State Diagram

9. Project Schedule

The project schedule is illustrated on the following Gantt chart.

10. Architecture Design

a. User Information Security

In the Login Page the users will enter their username and password. The system will check the username and password from the database. The passwords of the users will be kept in the database in an encrypted form. Login Page and the other pages of the system will be implemented using PHP technology[1]. The passwords of the users will be encrypted and then kept in the database to ensure security. PHP provides its own built-in hashing functions for encryption. The different interfaces for different users will be provided by PHP. After the user logs in, the system will check the user type and display the corresponding interface.

User passwords must be stored safely in the database. So, we must store password encrypted in database. We use MD5 function. "The MD5 message-digest algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly. MD5 function is also defined in php.[14]

b. Optimum Train Suggestion System

In the Home Page the passengers will plan their travels. They will search for the possible paths for their desired destination stations, departure stations and date. After selecting the departure and destination stations, 3 more drop down menus will appear. The user will choose the date and the train type. After the stations chosen, the system will show new drop down menus with the help of JavaScript[2] technology. The OTTS (Optimum Train Suggestion System) will determine the most suitable selections with the help of a minimum time/cost finding algorithm. The user will be able to sort the results according to travel time, cost, train type or number of changes. The system will do this by adding a final database query to sort these results according to the given entity. The result can be saved by the users to the personal page. The result appears as a link on the personal page. It will be kept in the user table of the database.

c. Intelligent Machinist Assignment

The machinists will check their daily schedule from Home Page. Intelligent Machinist Assignment will assign machinist to appropriate machines. The Brevets of machinists will be kept in the database. The system will match the machines with the machinist having the desired brevet and then choose one machinist to assign that machine.

d. Personal Information

In the Personal Page the users will be able to update their personal information. The system will fetch the information of the users from the database and fill the form with corresponding information. After editing, the system will update the information of

users in the database. And also travel history of users will be kept in the database and can be shown from this page.

e. Live Train Monitoring

In the Live Train Monitoring Page the users will be able to keep track of the trains on the map of the railways. This page will be implemented by using JSP technology[3]. It ensures to use Java functions and facilities in the web page. Since the main system functions will be implemented in Java, JSP is a better solution and more suitable for us to use. The map for the Live Train Monitoring will be drawn using Autocad Map 3D[4] and then it will be processed by some Java functions with GPS data. Java program will mark the positions of the trains on the map looking at the GPS data. The resulting map will have the positions of the trains and stations and it will be shown in the Live Train Monitoring Page. All of the users will be able to see the positions of the trains and also some information about the trains. The position and the other information will be updated periodically. The system will take the new coordinates and other information from GPS and update the result.

The machinists will also see the information and position of their own train and the other trains on the same line. On the machinist panel, the speed limit and the distance to the next station will also be shown. The speed limits for specific areas will be stored in the database. The system will check the current speed limit from the database with the position information and it will calculate the distance to the next station.

f. Emergency Situations

If an emergency situation occurs, the machinists will be able to push a button to warn the admin about the situation and send critical information to the admin. If a signal is received from the accelerometer, an emergency signal will be sent to admin and nearest control point automatically. The system on the machinist panel will process the signal coming from the accelerometer and send emergency signal.

g. Railway Map

The reason a new railway map is drawn with Autocad Map 3D is to see the system features and test the system more effectively enough on a suitable map. The railways in Turkey are not very complex and complicated. For every travel between cities there is only one possible path. We want our system to be useful in the future and it will be adaptable to any railway system. So, the current railway map of Turkey is not enough for our project. With this map the OTSS (Optimum Train Suggestion System) becomes useless. So, we will draw our own railway map and simulate the system on this map. Autocad Map 3D will be used to draw a new railway map. The map will only cover some part of Turkey and it will have many connections. It will be possible to select at least two different options from one city to another. Smaller stations will be connected to big cities with one railway and slower trains will be used.

h. Contact Information

In the Contact Page the users will be able to see the addresses, e-mails, phone number of stations, points to buy tickets, call centers. The user will be able to see the position of these places on a map. Google Maps[5] gadget will be used at this point.

Google Maps is a basic web mapping service application and technology provided by Google. It offers street maps, a route planner for traveling by foot, car, or public transport and an urban business locator for numerous countries around the world. There is only one problem with Google Maps. Since Google Maps is only free of charge for non-commercial use, it limits the number of requests in a certain time interval for a user. But since the gadget is not directly system related, the limit will only apply to the system users.

The users will be able to send e-mail directly by clicking on the e-mail addresses. For this purpose there is no need to use any extra application. With some simple HTML code this can be done.

i. Knowledge Database

In the Help Page the users will be able to search in a database of topics from database called Knowledge Database. It's mainly an issue tracking system. An issue tracking system is a computer software package that manages and maintains lists of issues, as needed by an organization. Issue tracking systems are commonly used in an organization's customer support call center to create, update, and resolve reported customer issues, or even issues reported by that organization's other employees. An issue tracking system often also contains a knowledge base containing information on each customer, resolutions to common problems, and other such data.[9]

Our system will provide passengers a knowledge database which allows them to solve their problem without calling the TCDD staff or sending a mail to them. If they can't find any solution, they will create a ticket. These tickets must be managed in a professional   way.

For this purpose we want to use an issue tracking system. With this system the administrators will be able to get new tickets from the passengers, assign them to the related persons and escalate them to the more experienced administrators. In this system they will also have a time restriction. All tickets will be solved in a particular time interval.

Instead of implementing such a complicated issue tracking system, we will integrate the application "JTrac" to our system. JTrac is an open source and highly customizable issue-tracking web-application written in Java. JTrac has many of the features of a standard issue-tracking application such as support for file-attachments and e-mail integration that we would like to integrate to our system. JTrac offers many customization options, especially in the areas of workflow and field-level permissions. We will take its most useful and basic parts ant let them work in our background environment. [10]

j. Database

As Database Management System MySQL[6] will be used. It's an open source software licensed under the terms of GNU GPL.

k. Webpage Design

We will need Apache Tomcat[7] as a servlet container to work with JSP pages. This is an open source software developed by Apache Software Foundation. Apache Tomcat provides a pure Java HTTP web server environment for Java code to run.

For the search in the web site we will use Google search. With some HTML code[8], the search string is sent to Google from the web page and the search scope is minimized to the current web site only.

11.<u>References</u>

[1].The PHPGroup (2001, dec 11). *Php hypertext preprocessor*. Retrieved from
http://php.net/index.php

[2]W3Schools,.(2001, Jan 16). *Javascript tutorial*. Retrieved from
http://www.w3schools.com/js/default.asp

[3] SunMicrosystems. (2002, March 3). *Java server pages technology*. Retrieved from
http://java.sun.com/products/jsp/

[4] NetBeans Foundation. (2003). Welcome to netbeans. Retrieved from http://netbeans.org/

[5] Google, . (2008). *Googel maps api*. Retrieved from http://code.google.com/intl/tr-TR/apis/maps/

[6]MYSQL Group, . (1999, Jan 27). *Mysql open source database*. Retrieved from
http://www.mysql.com//jsp/

[7] Apache Tomcat, . (2000). *Apache tomcat Welcome*. Retrieved from http://tomcat.apache.org/

[8] Taylor, Dave. (2002). *How can I add a search box*. Retrieved from
http://www.askdavetaylor.com/how_can_i_add_a_google_search_box_to_my_web_site.html

[9] Gnatt Project Team, . (2003). *Gnatt project home*. Retrieved from http://www.ganttproject.biz/

[10] Wikipedia, . (2008). *Issue tracking system*. Retrieved from
http://en.wikipedia.org/wiki/Issue_tracking_system

[11] JTrac. (2008). *Jtrac features*. Retrieved from
http://www.jtrac.info/doc/html/features.html/Issue_tracking_system

[12] Apache Software Foundation, . (2009). Welcome to apache foundation. Retrieved from
http://www.apache.org/

[13] AutoDesk Foudation. (2001). Autodesk home. Retrieved from http://usa.autodesk.com/

[14] W3 Schools. (2000). PHP md5() function. Retrieved from
http://www.w3schools.com/php/func_string_md5.asp