12306 Requirement Document

Content

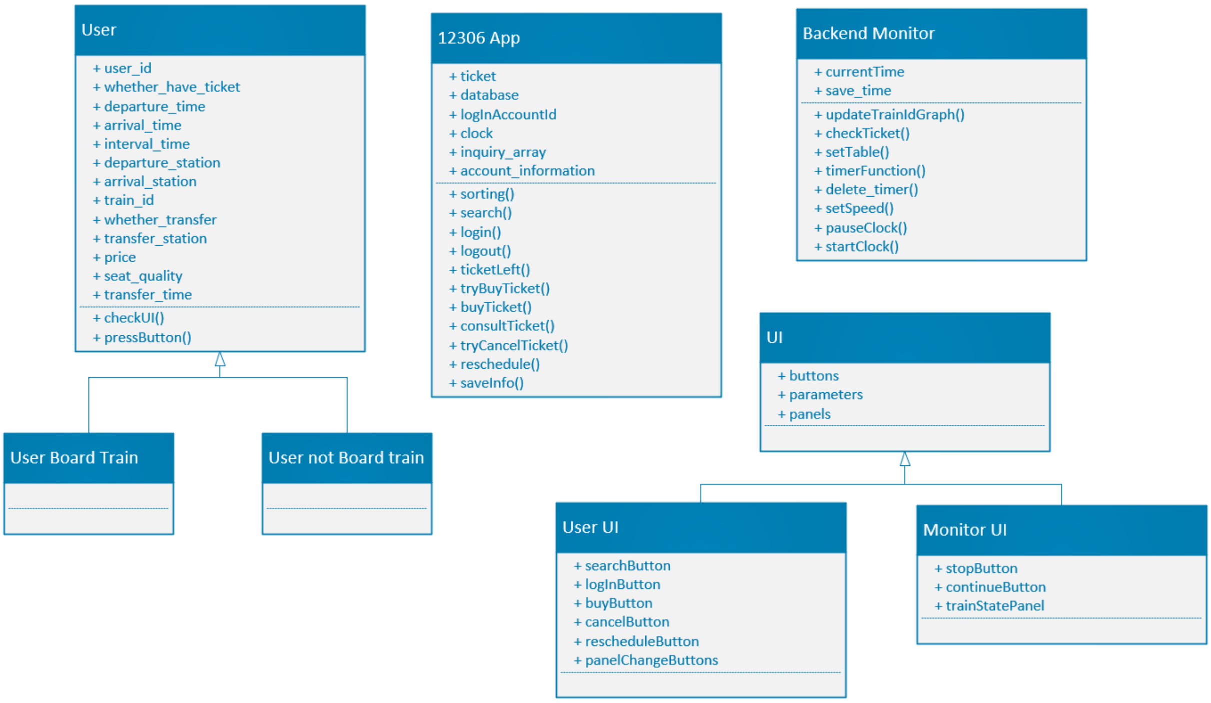
System Objective

In this project, we are developing a software that simulates the real world 12306 train ticket system in the scope of two railways as well as a interface to control system time and observe train state.

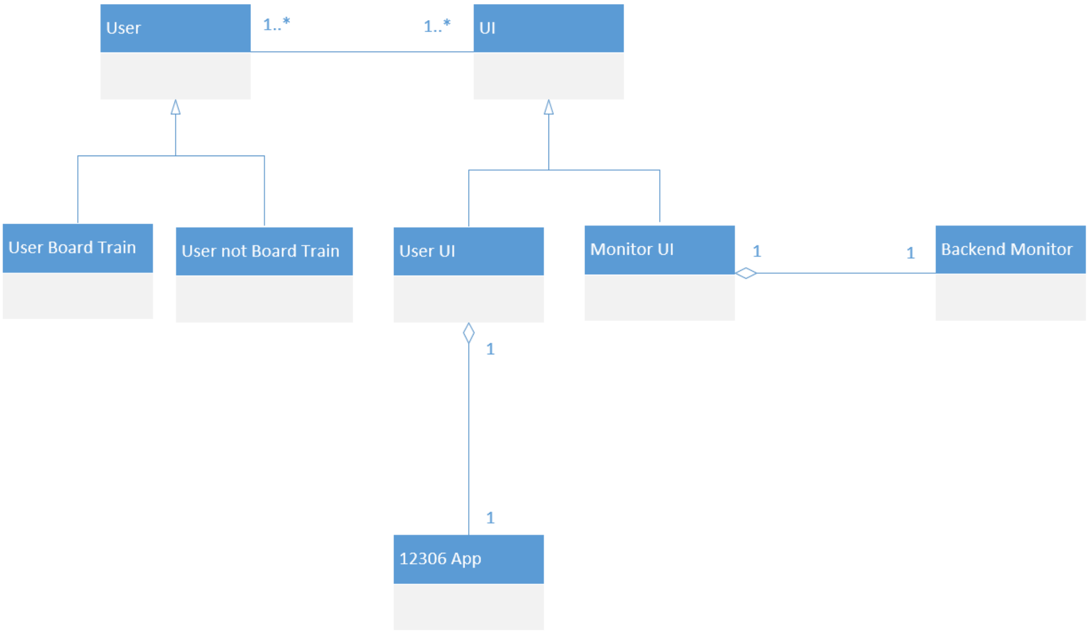
This software offers user the chance to experience buying, canceling, rescheduling train ticket and watch the backend changes as well as train state changes. User can also competitively buy the ticket in multi terminals, and operate on their tickets promptly.

Domain Analysis

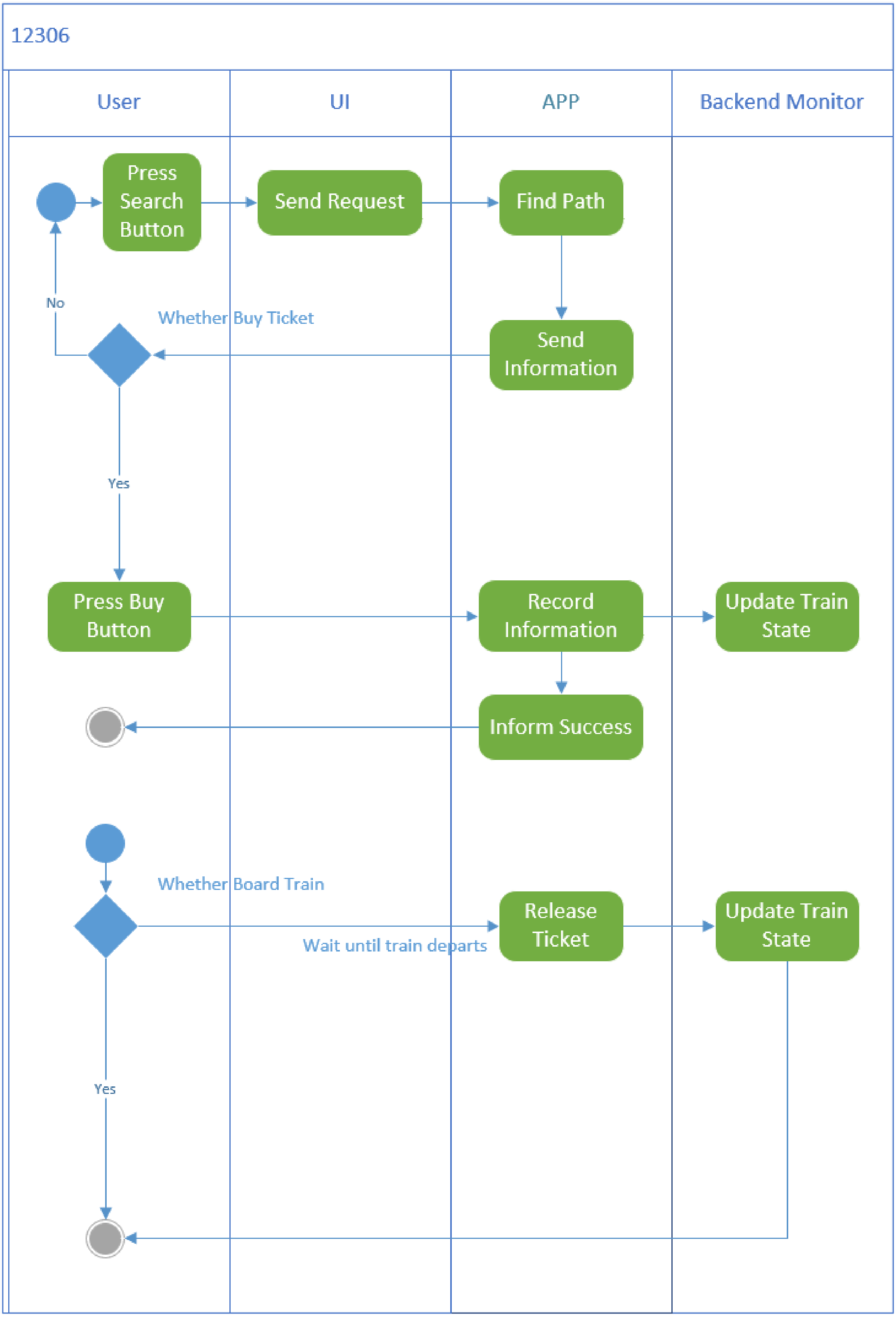
The participants of activities of the system can be categorized into three groups: user, 12306 app and system backend monitor.

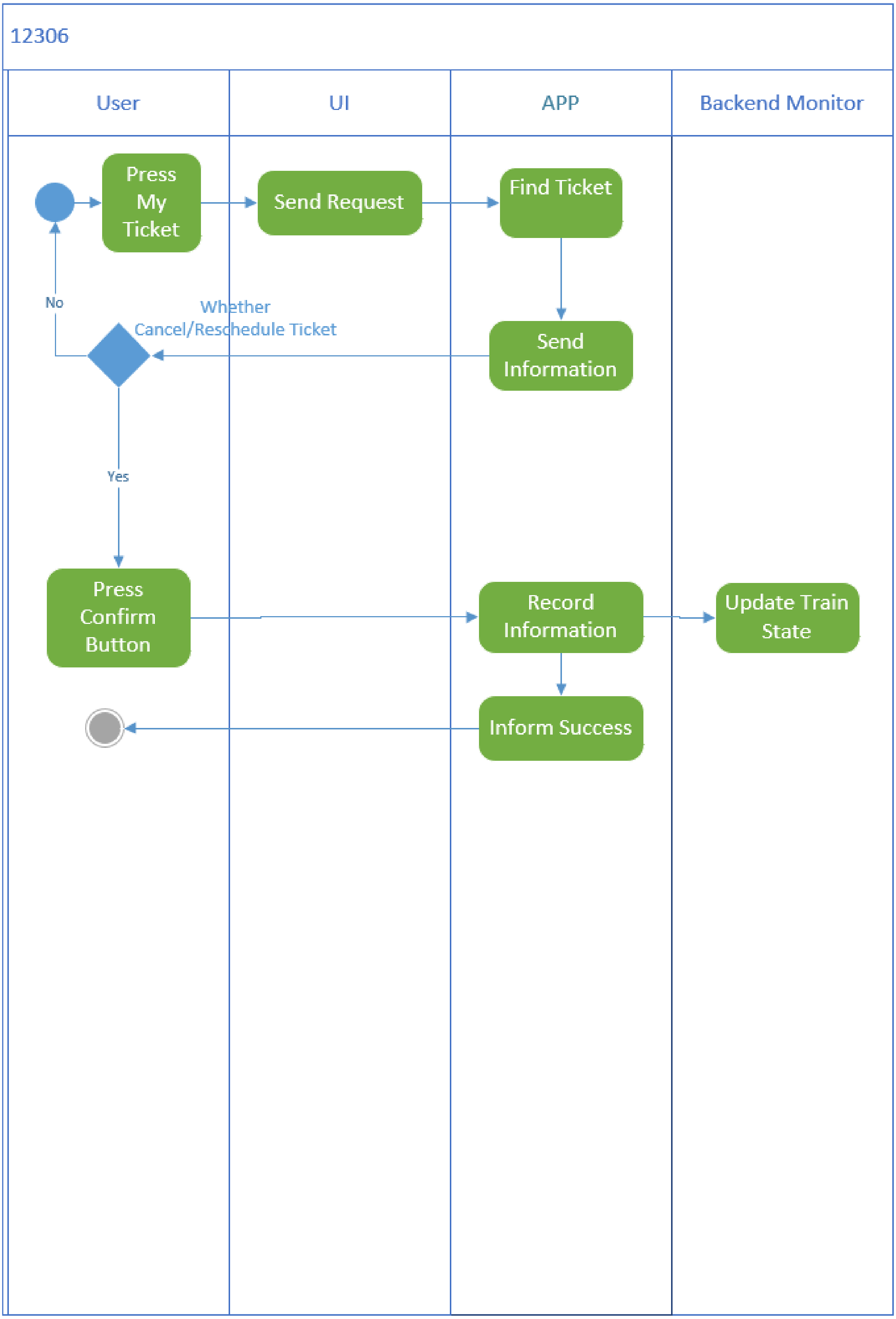


The relationship among different participants are shown as follows:



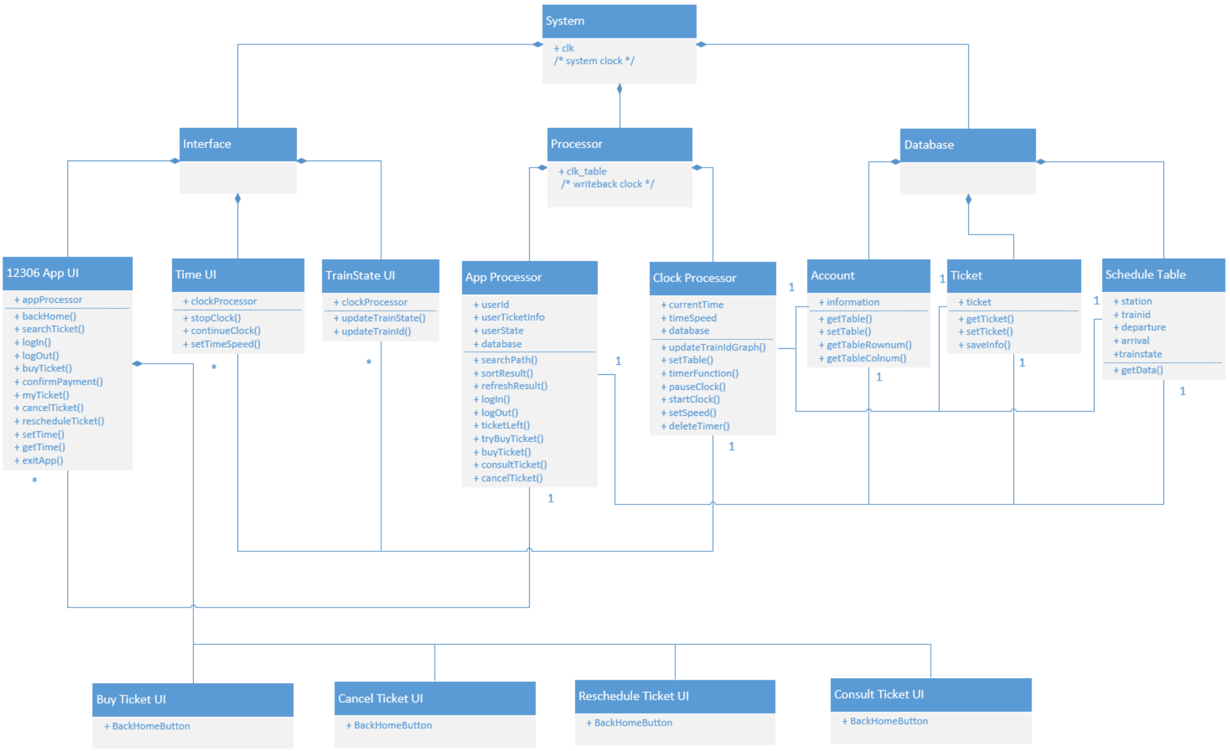
Below is the sequence of events for users use this system:


12306
User
UI
APP
Backend Monitor
Press Search Button
Send Request
Find Path
No
Whether Buy Ticket
Send Information
Yes
Press Buy Button
Record Information
Update Train State
Inform Success
Whether Board Train
Release Ticket
Wait until train departs
Update Train State
Yes


12306
User
UI
APP
Backend Monitor
Press My Ticket
Send Request
Find Ticket
Send Information
Whether Cancel/Reschedule Ticket
No
Yes
Press Confirm Button
Record Information
Update Train State
Inform Success

System Architecture

From the information above, we will design a software system that simulates the transaction process: Build the 12306 and backend monitor interface, allow users to interact with interface.

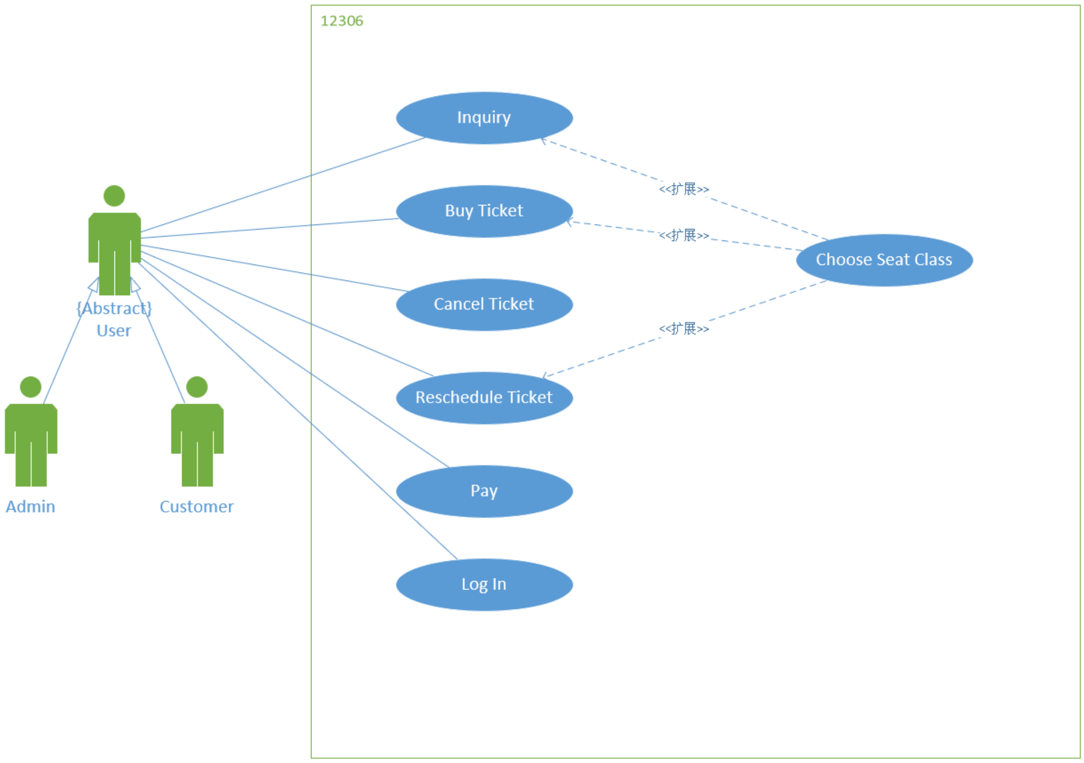


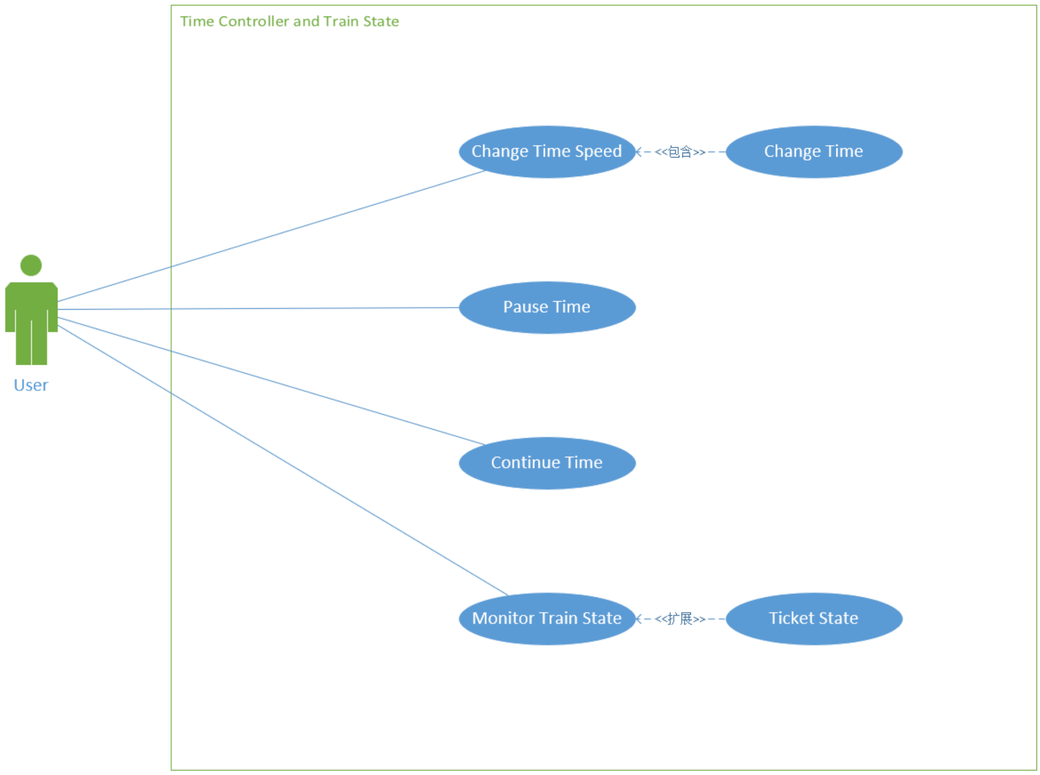
For further detail, in file structure there are `Account.m` and `Consult.m` respectively for deal with account and inquiry that go for App processor. `ConsultResult.m` is a non-handle class used as a struct. `Ticket.m` and `Database.m` are for system database. `Clock.m` is for system time calculations. Finally, `view.mlapp` is system UI and other `.csv` file is for data storage and will be written back from class every 2 seconds and after closing the app.

```
account.csv
Account.m
Clock.m
Consult.m
ConsultResult.m
DataBase.m
main.m
railway.png
savetime.csv
schedule.csv
ticket.csv
Ticket.m
ticket_save.csv
view.mlapp
```

Use Case

According to the objective above, the system should achieve the following use case.





R1 12306 APP

R1.1 User can search routines on main page

- R1.1.1 User can see current date.
- R1.1.2 User can choose sorting way, departure and arrival stations.
- R1.1.3 User can search available routines.
- R1.1.4 User can go to mine to look for informations.

R1.2 User can log in his/her account

- R1.2.1 User can choose which account he/she will log in.
- R1.2.2 User can log in the account he/she type in.
- R1.2.3 User can choose to interrupt login and go back.

R1.3 User can see Mine information after login

- R1.3.1 User can choose to log out.
- R1.3.2 User can see his/her ticket if ticket exists.
- R1.3.3 User can go back to front page.

R1.4 User can search for possible routine

- R1.4.1 User can interrupt the purchase.
- R1.4.2 User can buy the ticket he/she choose.

R1.5 User can pay for the ticket

- R1.5.1 User can choose the seat quality he wants.
- R1.5.2 User can interrupt payment.
- R1.5.3 User have a 15 minutes time limit on payment.

R1.6 User can do operations on his ticket

- R1.6.1 User can go back to front page.
- R1.6.2 User can reschedule his ticket.
- R1.6.3 User can cancel his ticket.
- R1.6.4 User can board/unboard the train.

R1.7 User can reschedule ticket.

- R1.7.1 User can reschedule ticket when he/she has a ticket.
- R1.7.2 User can interrupt thr reschedule process.
- R1.7.3 User can choose which ticket he/she wants to reschedule to.
- R1.7.4 User can choose seat class when rescheduling.

R2 Time Controller

R2.1 User can operate on system time.

- R2.1.1 User can type in the editfield and set timespeed.
- R2.1.2 User can pause the time.
- R2.1.3 User can continue the time.

R3 Train State Monitor

R3.1 User can see train location and seat occupation.

- R3.1.1 User can see train location and seat occupation move according to system time.
- R3.1.2 User can see the same time and state when restarting the app.

R4 User can buy ticket concurrently

R4.1 User can buy ticket under other user's effection.

- R4.1.1 Others buying ticket should effect all users' left ticket at the same time.