CMSC 345

Software Design and Development

UMBC-CMSC 447 Section 2 Team 4
Team Awesome

Planes for Hire

System Design Document

Client:
John Winder

Member:
Sundar Sekar <ssekar1@umbc.edu>
Tam Tran <tamtran1@umbc.edu>
William Cahill <wcahill1@umbc.edu>
Roberto Melgar <rmelgar1@umbc.edu>
Du Nguyen <du2@umbc.edu>

2/18/2015

Planes for Hire
System Design Document

Table of Contents

1. Introduction

1.1 Purpose of This Document

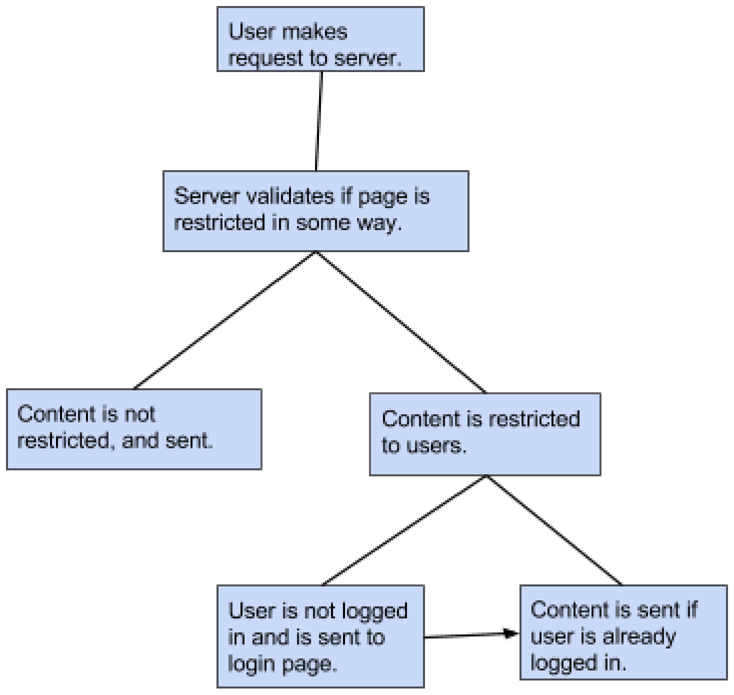
The purpose of this document is to describe the design of the Planes for Hire application. Key topics covered in this document include descriptions of high level user interface design layouts, lower level design, and description of persistence data implementation. A small amount of information will be added for Spiral 3.

1.2 References

1. Refsnes Data (1999-2015). w3schools.com. Retrieved from "http://www.w3schools.com/googleapi/"
2. The PH Group (2001-2015). PHP. Retrieved from "http://php.net/"
3. QuinStreet Inc (2015). SQLCourse.com. Retrieved from "http://www.sqlcourse.com/"
4. The Planes for Hire System Requirements Document

2. System Architecture

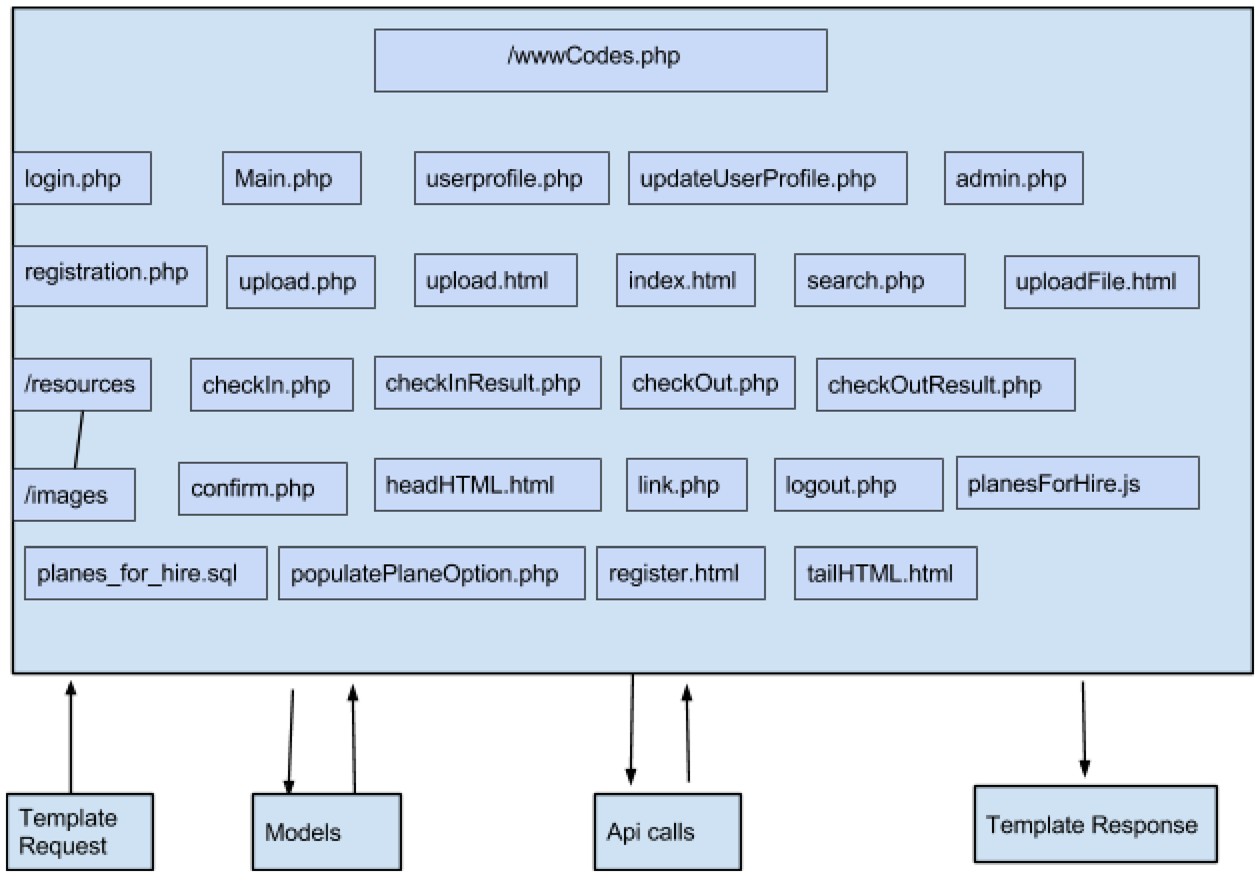
2.1 Architectural Design


User makes request to server.
Server validates if page is restricted in some way.
Content is not restricted, and sent.
Content is restricted to users.
User is not logged in and is sent to login page.
Content is sent if user is already logged in.

Web application will be stored on UMBC's gl server. The server is a linux-based server. Our developers and customer can easily communicate with it by using an application such as secure shell or putty. This simplifies the process of adding, deleting, modifying, dragging and dropping files to and from the server. GL server uses Apache as the application that simplifies file access, we will use PHP, html, javascript, and ajax as the backend code, and PHPMyAdmin as the SQL database.

We know that the front end code will be written in html and javascript, which will be the the code that is executed in the user's browser. Server-side instructions will not be included. We know that our PHP documents are used to execute the majority of the backend server code, and send generated html documents to the user while stripping server-side instructions. We know that our PHP code will communicate with the MySQL database, and that the database stores a collection of tables We know that the Javascript, CSS Bootstrap wil used extensively in order to add interactivity to a webpage's gui elements.

2.2 Decomposition Description

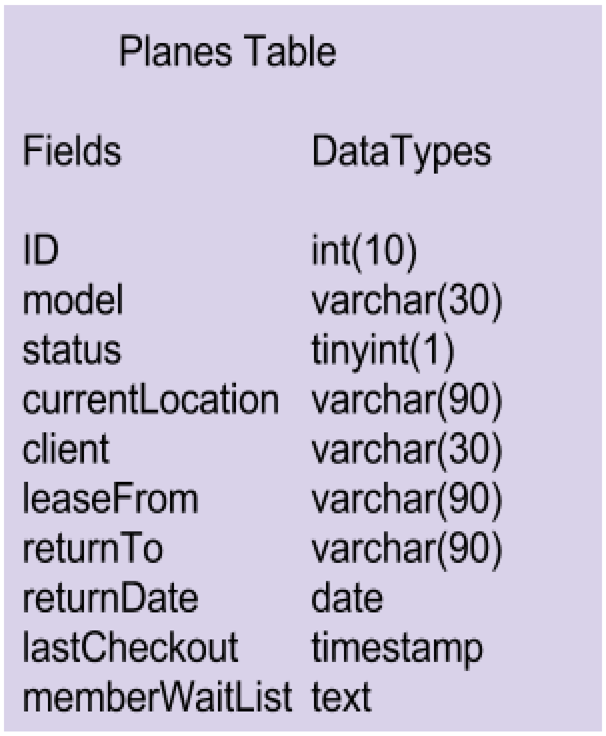


Each page behaves as a module that is imported to the main page. This allows each of us to work on separate files, and have it be plugged into the main one with ease. Our coding style will lean towards functional and procedural, meaning that there will be no try-catch statements, inheritance, etc. Errors instead will be checked using if-statements after an operation has been performed. There will be php files specifically designed as modules that will contain utility functions such as functions for creating, removing, and maintaining accounts.

3. **Persistent Data Design**

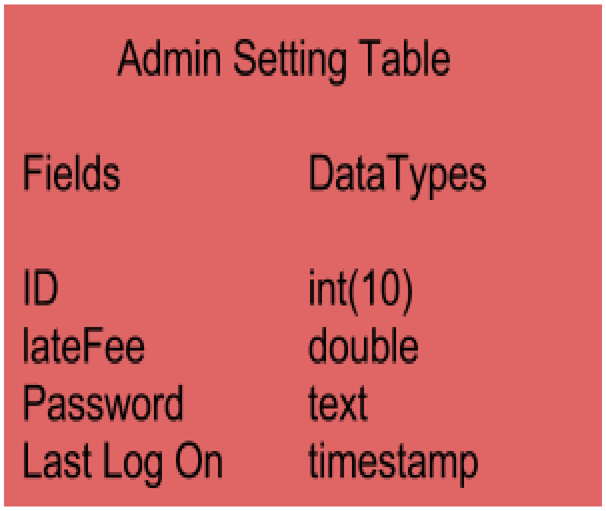
3.1 Database Descriptions

All tables are kept in a phpMyAdmin SQL database named planes_for_hire. This database was created by Tam. All other members were able to create their databases by importing the planes_for_hire.sql. the use and creation of this database was free. The tables that were created are the customer table, airport locations, administration settings, and planes for each users. The details for each table is described below.



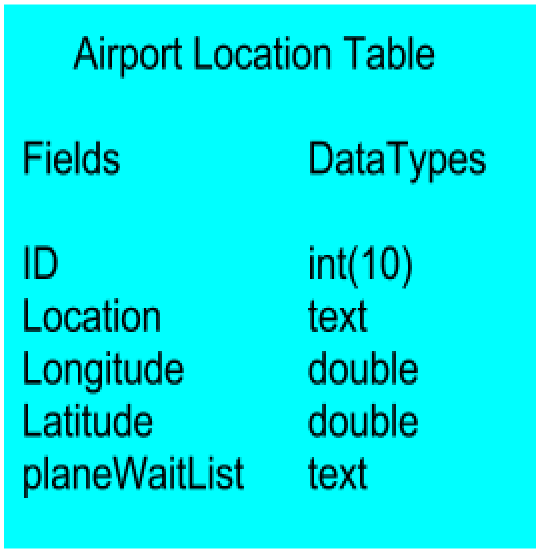
Planes Table

Fields	DataTypes
ID	int(10)
model	varchar(30)
status	tinyint(1)
currentLocation	varchar(90)
client	varchar(30)
leaseFrom	varchar(90)
returnTo	varchar(90)
returnDate	date
lastCheckout	timestamp
memberWaitList	text



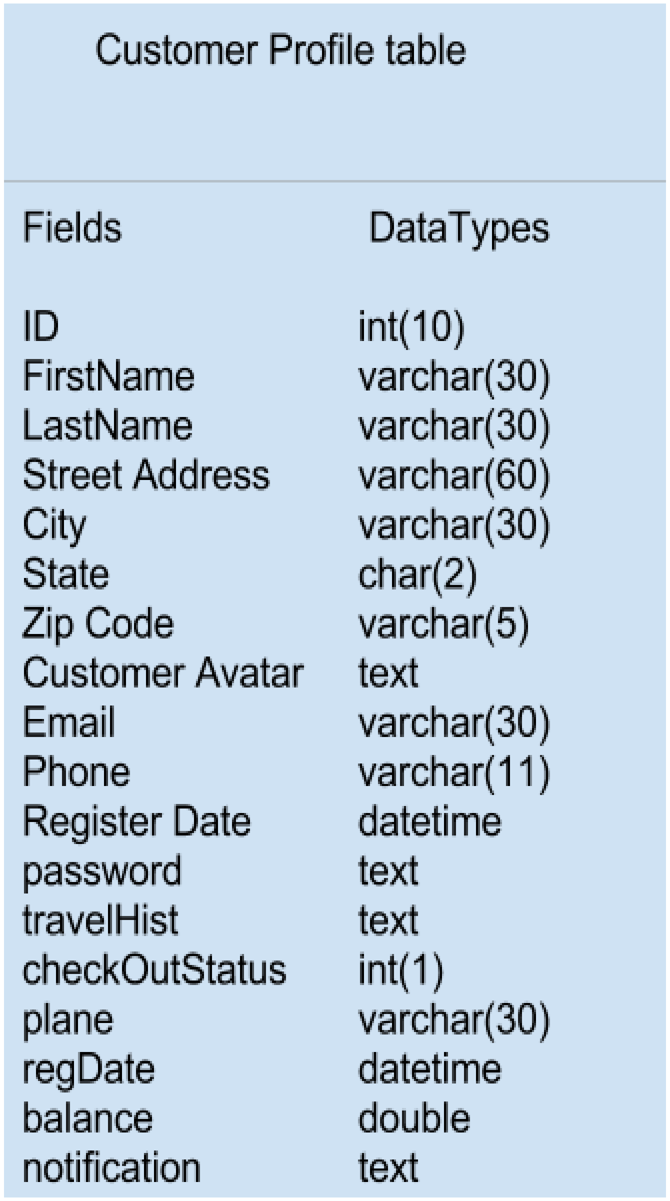
Admin Setting Table

Fields	DataTypes
ID	int(10)
lateFee	double
Password	text
Last Log On	timestamp



Airport Location Table

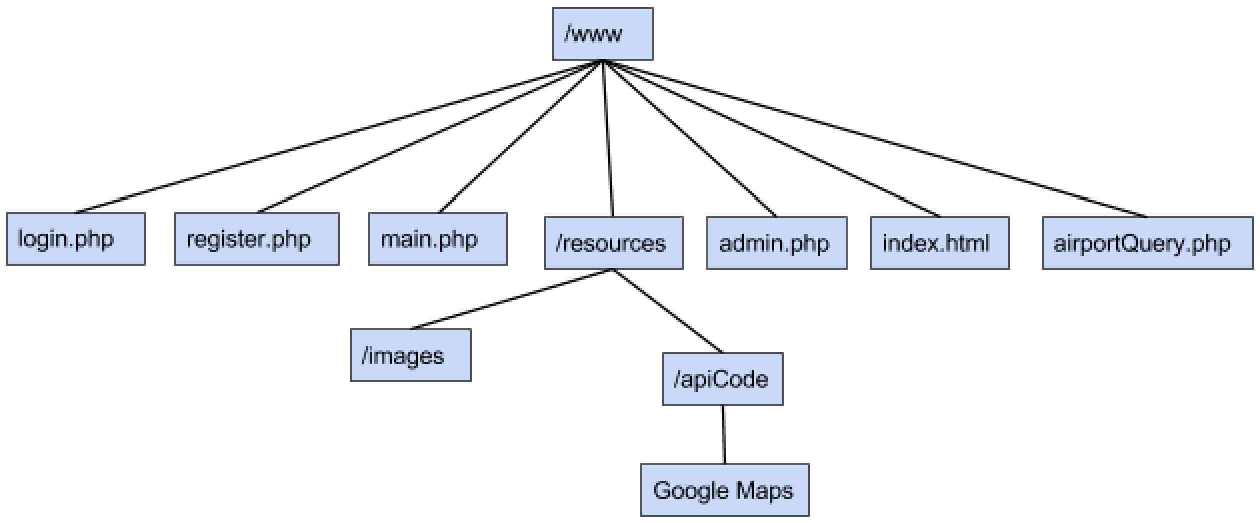
Fields	DataTypes
ID	int(10)
Location	text
Longitude	double
Latitude	double
planeWaitList	text



Customer Profile table

Fields	DataTypes
ID	int(10)
FirstName	varchar(30)
LastName	varchar(30)
Street Address	varchar(60)
City	varchar(30)
State	char(2)
Zip Code	varchar(5)
Customer Avatar	text
Email	varchar(30)
Phone	varchar(11)
Register Date	datetime
password	text
travelHist	text
checkOutStatus	int(1)
plane	varchar(30)
regDate	datetime
balance	double
notification	text

3.2 File Descriptions

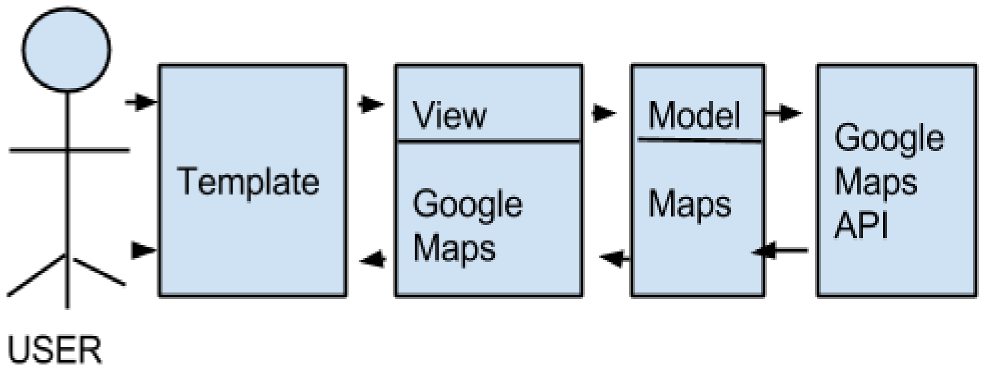


Root directory of website is the /www folder. Index.html is the bootstrap that will load main.php. Main.php contains majority of the application's content, and loads images and api code from the resources directory. Login.php is a modular file that is called from main.php to retrieve a user's profile from the database, which is directed back to main.php. This is also the same for airportQuery.php. Register.php is a separate page that allows a users to create an account, which will be pushed into the database, and then the user will be redirected back to main.php. The user is only directed to register.php from main.php. Every database access is directed back to the main, and the result is shown in main. Search queries will invoked from main which calls airportQuery.php to access the database which will return a list airport candidates into main. Admin.php is a page that is only accessible to administrative users, which is able to display all users in the database, and all planes they have checked in and checked out currently, and in the past. Admin page is also capable of adding planes and airports to the database, which will make them available to users.

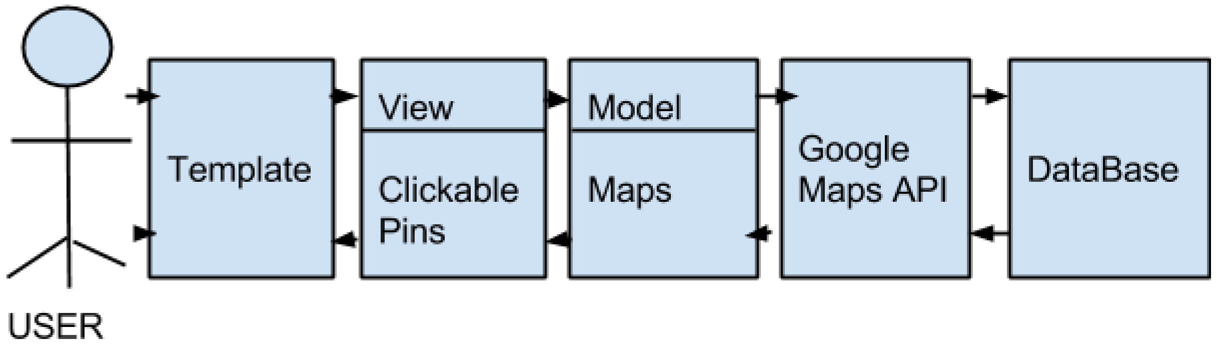
4. Requirements Matrix

Please refer to the System Requirements Specification for details regarding the corresponding use cases.

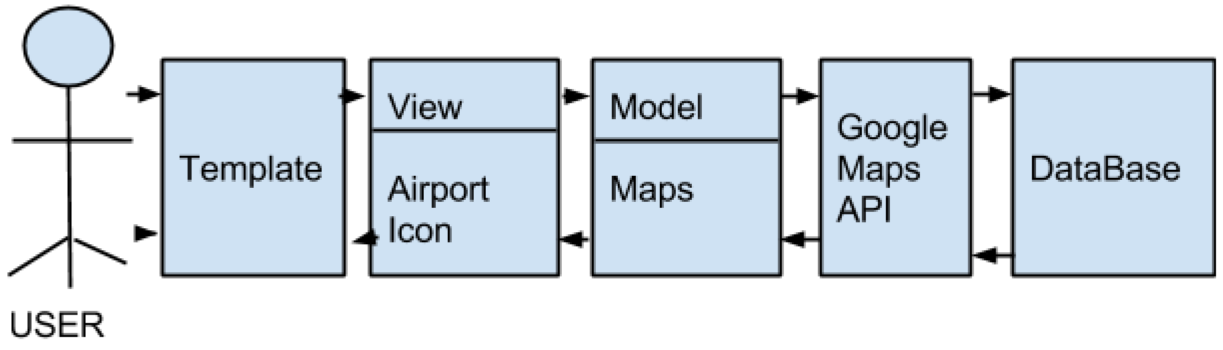
Google Maps: Use Case # 1



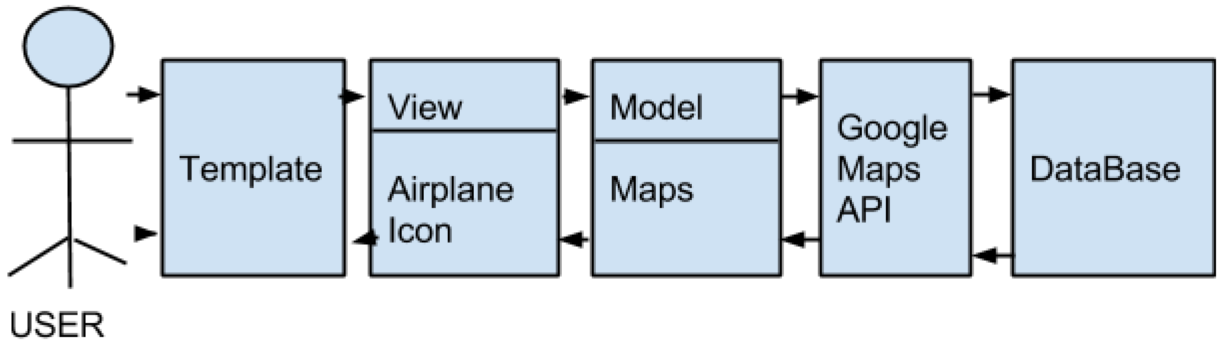
Clickable Pins: Use Case # 2



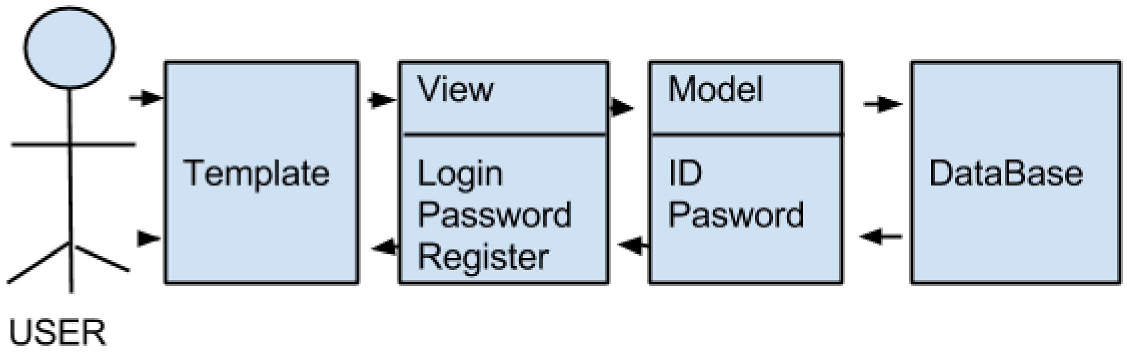
Airports: Use Case # 3



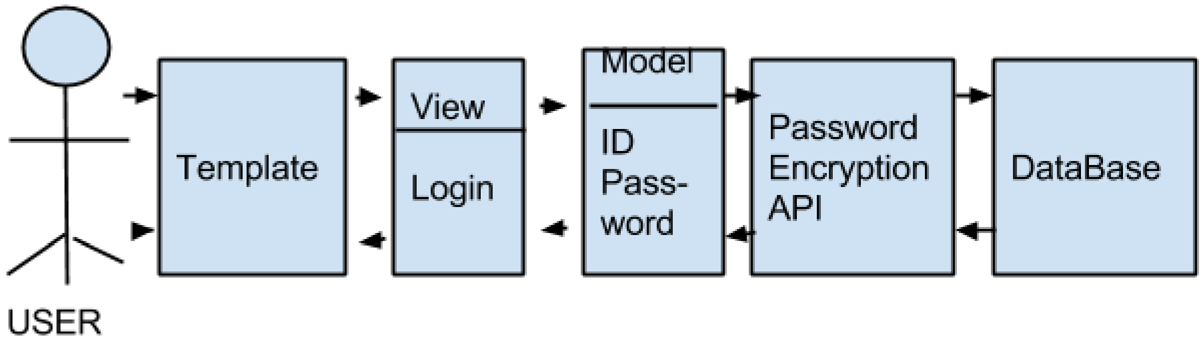
Airplanes: Use Case # 4



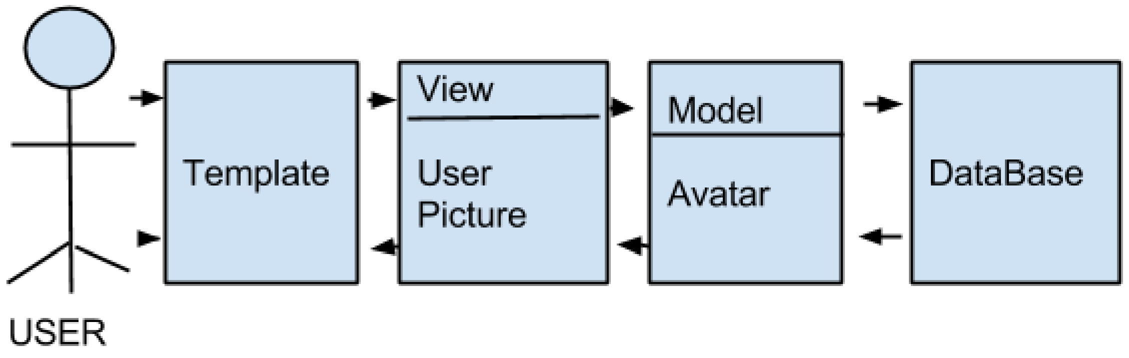
User Login: Use Case # 5



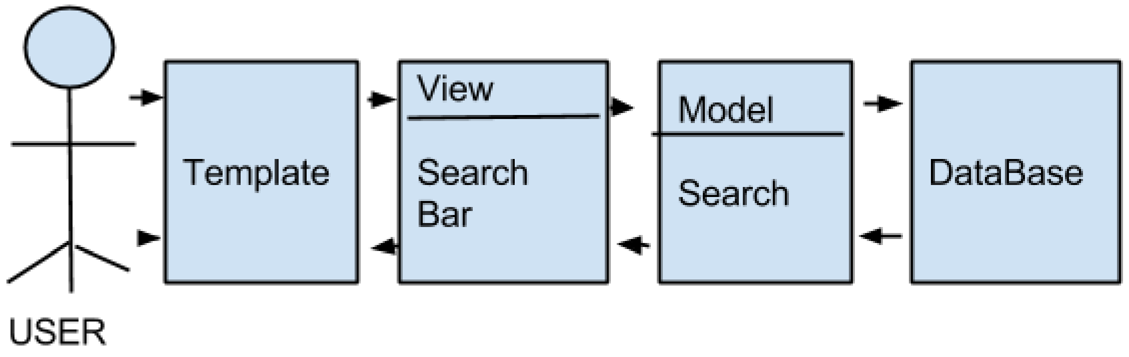
Account Protection: Use Case # 6



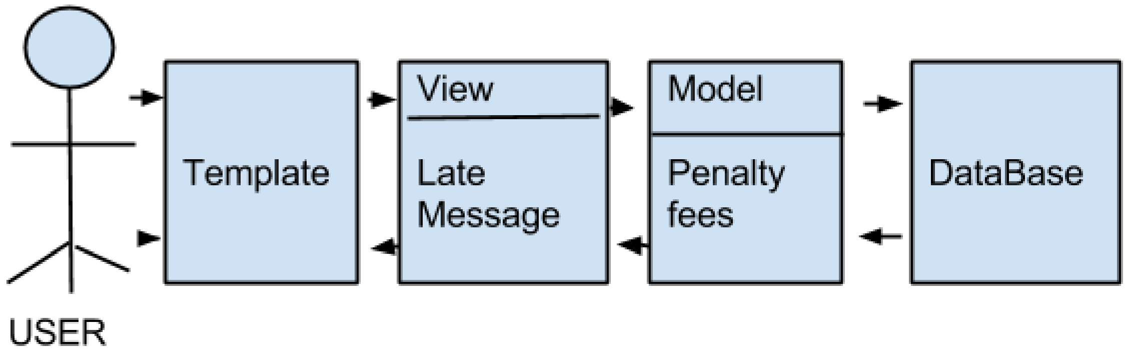
Avatar: Use Case # 7



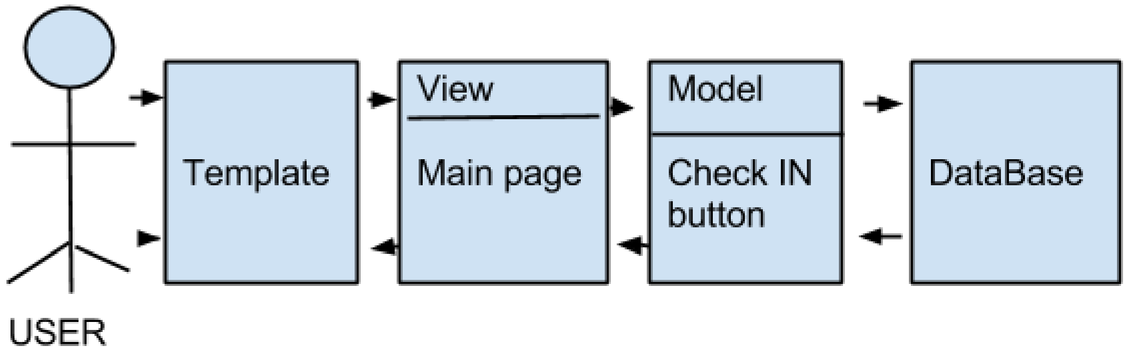
Search: Use Case # 8



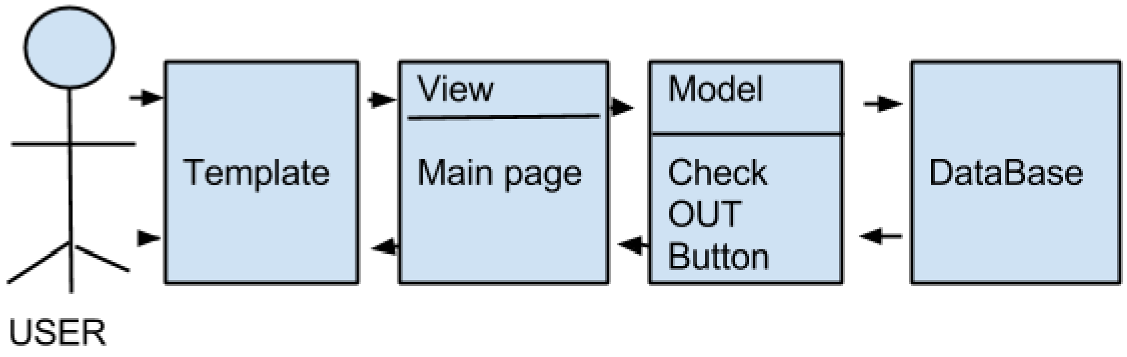
Late Penalties: Use Case # 9



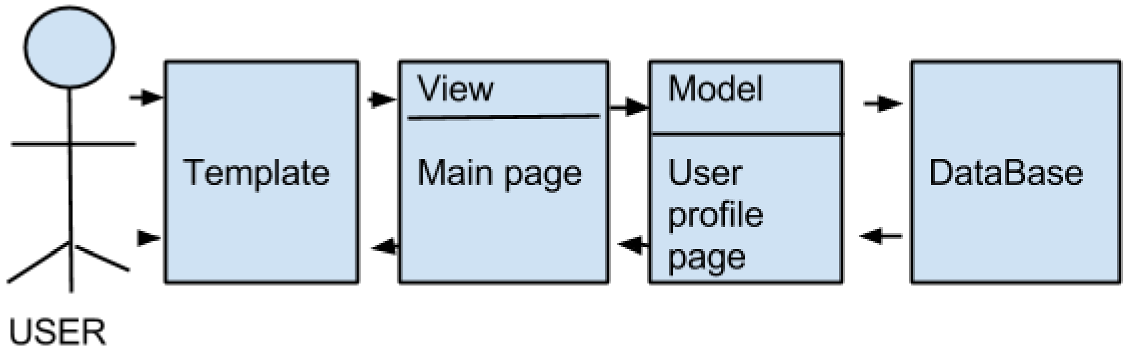
Check IN: Use Case # 10



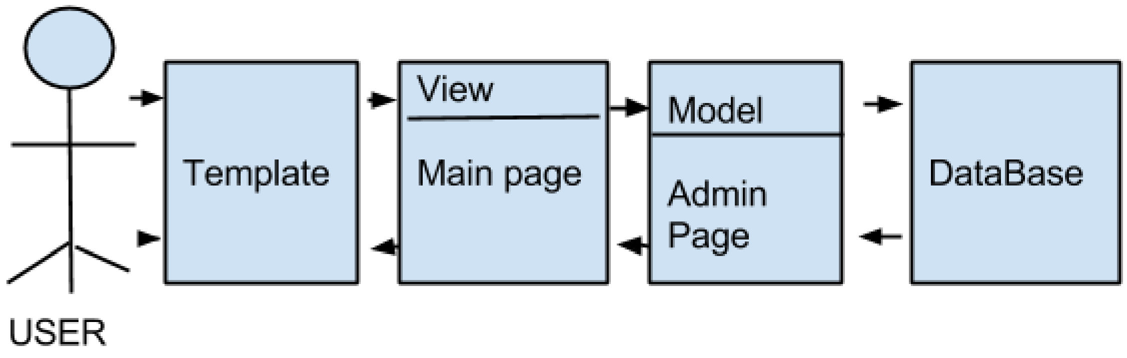
Check OUT: Use Case # 11



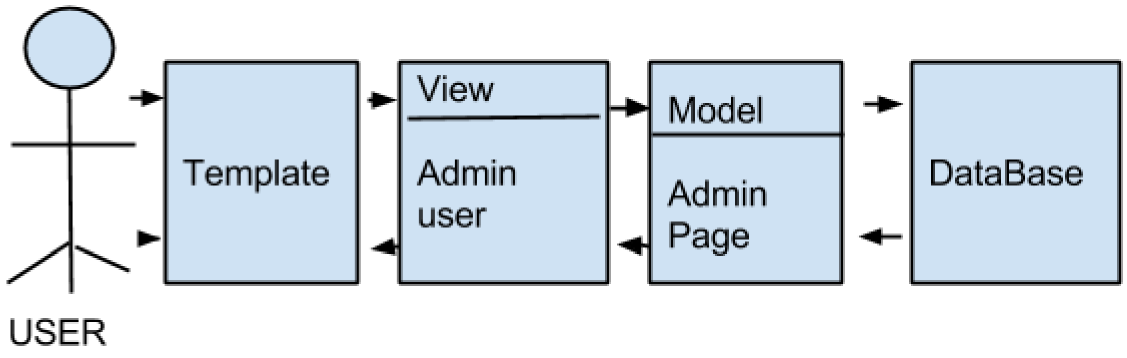
User profile page: Use Case # 12



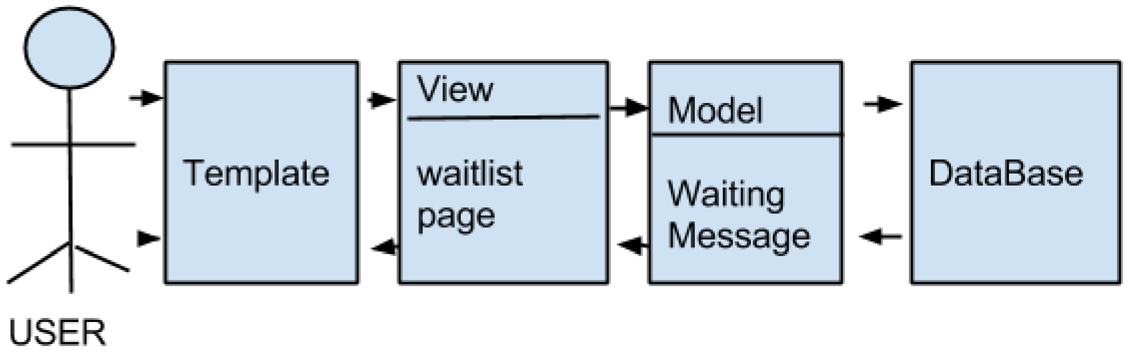
Administrative User: Use Case # 13



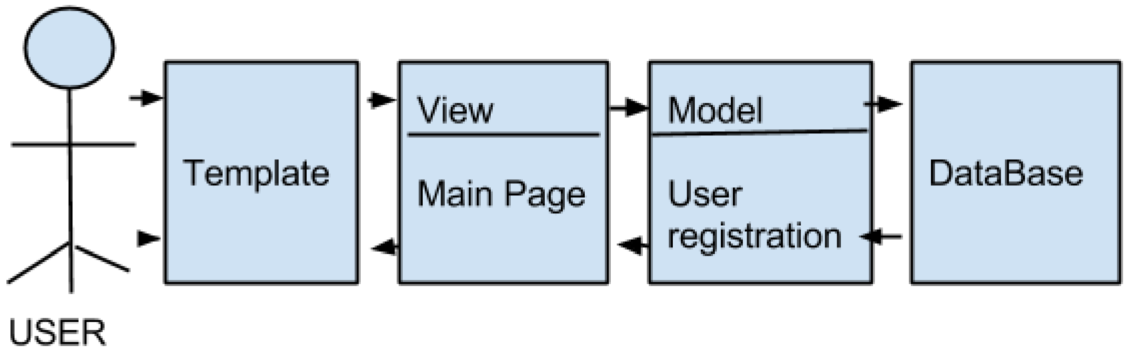
Administrative page: Use Case # 14



Waiting list: Use Case # 15



User registration: Use Case # 16



Use Cases	Files/Resources Used	How are they used?
Use Case #1	Main.php, resources/images/pin.png, headHTML.html, tailHTML.html planesForHire.js	Main.php is loaded, bringing in headHTML, loading google maps API, displays a map, displays a list of planes to non-users, allows users to check in and out planes, and brings in tailHTML to finish it off.
Use Case #2	resource/images/pin.png planesForHire.js	When one pin is clicked, the map centers on it and fills in check in/out fields. When two are clicked, a line is drawn between the two airports. planesForHire.js creates the pins with an image.
Use Case #3	resource/images/pin.png planesForHire.js	Pins populate the map, and use the pin image stored in resources/images. planesForHire.js create pins with the image.
Use Case #4	Database Main.php	Planes are populated from the database tables to main.php.
Use Case #5	Database registration.php	User registers an account by inserting information. This will create a form that will be sent to

Use Case #6	N/A	Will be implemented for Spiral 3
Use Case #7	userprofile.php registration.php picsUpload/ upload.php	User is able to upload an image for their profile picture. In upload.php, the code checks if the image is an actual image, if the size is ok, and if it is one of the acceptable formats. This includes, JPEG, JPG, GIF, and PNG.
Use Case #8	search.php main.php	User is able to search for airport locations.
Use Case #9	Database userprofile.php	User profile displays error message on their profile page if user is late for payment. Amount due is also displayed. Data regarding payments are stored in database.
Use Case #10	Database Main.php	When user checks in a plane, whatever plane is currently checked out by the user is freed and the user makes their payment. They are directed to checkInResult.php
Use Case #11	Database Main.php checkInResult.php	When user checks out a plane, all fields in form must be filled out. Once done, form data is sent to database including departing airport, arrival airport, and starting date.
Use Case #12	userprofile.php Database main.php	User is able to check their travel history log, update profile avatar, pay fees, and change their user information
Use Case #13	admin.php adminHeadHTML.html adminTailHTML.html	Admin user will be able to add new users, new airports, new airplanes, delete users or suspend users
Use Case #14	admin.php adminHeadHTML.html adminTailHTML.html	Travel history of users.

Use Case #15	N/A	Will be implemented for Spiral 3
Use Case #16	Database registration.php	User creates a profile by typing in the information in the bars. The code in registration.php takes all values typed in and assigns them to variables. Then the new user is stored into the database

Appendix A – Agreement Between Customer and Contractor

The customer agrees to a Planes For Hire web application with searching, google maps, and user accounts with security encryption capabilities. See System Requirements Specification for more information. Additional features will be provided in further development spirals.

When and if future changes to this document occur a drafted new document will be created. Both a hard and electronic copy of both versions will be presented to the client for review. Upon approval, the draft will be finalized and signed off by both parties.

Client
Name ____
Date ____
Sign ____

Team
Name ____
Date ____
Sign ____

Name ____
Date ____
Sign ____

Name ____
Date ____
Sign ____

Name ____
Date ____
Sign ____

Name ____
Date ____
Sign ____

Appendix B – Team Review Sign-off

This document has been collaboratively written by all members the team. Additionally, all team members have reviewed this document and agree on both the content and the format. Any disagreements or concerns are addressed in team comments below.

Team

Name ______________________________
Date ______________________________
Sign ______________________________
Comments ______________________________
____
____

Name ______________________________
Date ______________________________
Sign ______________________________
Comments ______________________________
____
____

Name ______________________________
Date ______________________________
Sign ______________________________
Comments ______________________________
____
____

Name ______________________________
Date ______________________________
Sign ______________________________
Comments ______________________________
____
____

Name ______________________________
Date ______________________________
Sign ______________________________
Comments ______________________________
____
____

Appendix C – Document Contributions

Tam and Andrew took the lead on this document. Andrew drew the architectural design and wrote two paragraphs for section 2.1. He also wrote the file description and drew the graph. Tam searched for the references. Tam drew most of the images for document description. Andrew, Sundar, and Tam worked together on the decomposition description. Tam and Sundar discussed how to design the tables for 3.1 while Du wrote the description. Sundar an Tam also collaborated on the Requirement Matrices. Sundar Sekar created all the diagrams in requirement matrices, and also wrote description for purpose. He also assembled, reviewed and edited the document. Roberto worked on the purpose of the document and the creation of the Appendices. He also deleted some of the template examples.

For Spiral 2, Sundar, Andrew, & Roberto updated doc.
For Spiral 3, Roberto updated tables in 3.1