

**Navy Project**

**GovAppStore Prototype**

**Software Requirements**

**CS337**

**Software Design**

**Prepared By:**

**Brandon Ung**

**Maximiliano Barragan**

**Andrew Aquino**

**Kevin Parton**

**Richard Thome**

**Faculty Advisor: Jiang Guo**

**October 24, 2015**

**SECTION……………………………………………………………………………………PAGE**

1. **Versions…………………………………………………………………………………………………………3**
2. **Introduction………………………………………………………………………………………………4**
3. **Overall Description……………………………………………………………………………5**
4. **External Interface Requirements……………………………………………6**
5. **User Interface Overview…………………………………………………………………6**
6. **System Features (Functional Requirements)………………13**
7. **MODELS……………………………………………………………………………………………………………14**
   1. **USE CASES**
   2. **STATE DIAGRAMS**
   3. **SEQUENCE DIAGRAMS**
   4. **DATA TREES**
   5. **ACTIVITY DIAGRAMS**
8. **Nonfunctional Requirements………………………………………………………39**

**Glossary**

**1 - Versions**

**A = Added M = Modified D = Deleted**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version #** | **Date** | **A, M, D** | **Description** | **Change Request Number** |
| **0.1** | **10/4/15** | **A** | **Working registration and login authentication** | **1** |
| **0.2** | **10/11/15** | **A** | **Lifted servers using sails.js** | **1** |
| **0.21** | **10/18/15** | **M** | **Sails server modified to run Polymer and JQuery** | **2** |

**2 - Introduction.**

**2.1. Software Purpose**

Cloud computing is internet-based computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources. The Navy is at the forefront of transitioning its science allocation to the cloud due to the advantages of (1) increasing flexibility of computing systems that can dynamically make adjustment and instantly meet the demand changes, and (2) more reliable computing services and decrease maintenance cost by supporting automatic software updates.

GovAppStore Prototype is a cloud computing based system designed to improve NAVSEA mobile apps delivery and updating. This project is similar to Google's AppStore or Apple's iTunes. It'll be a way to distribute government built mobile apps to government tablets/smart phones.

**2.2. Software Scope**

The target platform will be Microsoft Surface. Students will build the server side application that will host dummy test applications and client side GovAppStore app. The client side app will be able to connect to the server side and pull a list of applications available to download. If the application already exists on the client side tablet, it will check the version number to see if there's an update available or not.

**2.3. Software Perspective**

The user will find this application an easy way to search for popular or specifics apps found on the server. Users will be able to create their own accounts dedicated to this website. The account holds personal informations pertaining to the user as well as uploads and download history of applications. Users can search via the search bar to query specific items found on the database.

**3 - Overall Description**

**3.1 User Description**

The user of this application does not need any in-depth knowledge of computers. The only requirement in order to register for an account onto the application is an existing, current, email address.

**3.2 Operating Environment**

This application is developed to be able to run on any operable browsing program such as Internet Explorer, Firefox, Google Chrome, and Opera. The latest versions of each browsers is recommended for better access to the website’s interface and implementation.

**3.3 Design and Implementations**

The application utilizes node.js for back-end implementation of the application. HTML5, Polymer, and JQuery is primarily used for the front-end implementation of the application. Bootstraps is also implemented for the overall presentation of the application. Sails.js, a Model-View-Controller framework, will be incorporated with node.js, Polymer, and JQuery. Sails.js is used to immensely organize the structure of files and data.

**3.4 Databases**

The application uses MongoDB as one of the primary databases to store user information needed for the application. Such information will be the user’s name, email, password, download history, uploads and etc. MongoDB allows for easy reading and writing of large amounts of data.

**4 - External Interface Requirements**

**4.1 External Software Requirements**

This application requires a web browser such as Internet Explorer, Firefox, Google Chrome, and Opera. Latest version for each browser is recommended.

**4.2 External Hardware Requirements**

Basic computer setups (Monitor, keyboard, desktop)or laptop is required to work with this application.

**5- User Interface Overview**

The software application in terms of the client side aspect will be designed for the ease of the user experience. This is imperative for the application because users must get the most utility out of this application as possible. The visual aspect will need to be captivating in order to have users prefer this app-store site over the competing app-store sites such as the Apple Store and Google Play.

We will incorporate the visual aspects that Apple Store and Google Play implement such applications, photos of apps and a user interface that has vibrant colors in order to be aesthetically pleasing to the users of the application. Making the application more aesthetically pleasing is meant for attracting many users because there is nothing else that will really differentiate our application from Apple Store and Google Play since we will have the same usage, downloading and viewing applications on the website. The target demographic will greatly be assisted by the ease of use of this application. That is why it is imperative to make a very vibrant, welcoming, and easy-to-understand implementation in our interface to facilitate a euphoric and uplifting feeling when users use our application for downloading and finding useful applications for their work.

Additionally, many users will particularly enjoy the variety of apps from the application. Certain free applications will include advertisements unless they upgrade to purchasing the full app. This will give our users great accessibility to our application’s services and in return knowledge will spread around about this product and the users will increase exponentially. This exponential growth in users is also another reason why the interface will have to be very easy to use and straightforward. We also have to accommodate for non-tech savvy users and make an interface such as Facebook or any other social media application that has an easy to use and colorful interface that anyone could use.

When a user first enters into the site or application they will see popular applications that have gained popularity in the top app ranking. It will include a large photo of the app that is front and center of the interface and will have the app’s name. The revolving photos of the top ranking applications will consist of the top 5 applications. We mention this first because it is the first thing that the user will most likely see upon entering the webpage. Furthermore, once the user clicks on the photo, it will take them to the app’s information page. Of course for the downloading of apps feature will be available to them only if they register and make an account.

Now I will describe the navigation bar on top of the webpage. There will be one button that is labeled “Sign Up”. When the user clicks this button, it will take them to a web page that allows them to register for the application. There will be text fields, which they are required to fill out. The first two text fields will be for their first and last name. The next text field will be for them to provide a valid email address, which they will have to verify when we send them an email to the email address that they provide us with. The next two text fields will be for the user to create a password and to verify the password by entering it again. The password will be required to be at least 8 characters long and there will be a password strength indicator that tells the user how difficult their password would be to guess. The next text field will be for an additional email address for the user to provide for safety purposes and will be an optional field. There will be some security questions that the user can choose from and create their own answer in the event that they forget their password and at the very end of the process there will be a CAPTCHA text field where the user will have to type a message that is shown in order to determine whether a human is registering for a legitimate user account.

The next button on the navigation bar will be a “Login ” button. This will just take the user to a login page where there will be two text fields. One will be for the user’s email address and the other will be for the user’s password. Once both pieces of information are entered, the application will verify if the user email and password are in fact belonging to a valid user of the application and if it is then the user will have the application available to them and they can start downloading applications. If the user email and/or password is not valid then a message in red on the login window will read “Your email address and/or password is not valid, please try again.”.

The final button on the navigation bar will be a “Help” button. This will take the user to a web page full of Frequently Asked Questions (FAQS) by other users. It will also display an email address to one of our tech assistants that can assist the user with any technical issues.

Once the user has successfully logged in to his or her account they will go to the user page. There will be a section where they can see applications the website have recommended. If they click on a particular app’s page they will be redirected to that page with the applications information.

The user will also see suggested applications that will be generated based on their history. For example, if the user uses a lot of Productivity or Social Media applications, the recommended apps will include Word Document editors, facebook, twitter and many more. The user can specify on their profile what Genres of apps they enjoy most. Depending on their preferred genres, more suggestions will be generated.

There will be different ways to search applications on the user page. On the very top there will be a text field that is labeled search. A user can type in a app name, genre or any keyword in order to query the database so they can find an application that they can download. Whatever the user enters into the search box will be used in a query to find whatever objects in the database that contain keywords that are provided by the user.

Other methods of searching will consist of having a top selling applications, most popular applications, and most viewed applications. There will be a more general method of searching through selecting a genre. The genres that will be available to choose from will be the basic genres that would be found on other application websites. There will be Productivity, Tools, Streaming, eBooks, Games, Communication, Education and Social Media. We will add more genres if there are enough requests during the software testing stage when we offer the application to test subjects. These test subjects will be people that are selected from different regions of the country, different ethnic backgrounds, and different interests in general. This way we can have a good idea of how millions of people will like this application. If different people with different backgrounds like this application or if we accommodate their needs then it is more than likely that most people around the country will enjoy using it. This is a mere assumption and this application has yet to be tested and further prototypes are still to be developed.

Overall for all the web pages that will make up this app store application, we will be using Bootstrap and for more elaborate designs and styles we will use Cascading Style Sheets. Using Cascading Style Sheets and be very tedious and laborious and the interface could look very unpleasant therefore Bootstrap will be the go to method of making the user interface aesthetically pleasing. During the software testing process, we will conduct many surveys asking the test users what they would like to see changed in the interface. After we conduct these surveys, we will implement any feasible and popular suggestion for the user interface. Using Cascading Style Sheets will prove to very laborious, but Bootstrap cannot fulfill the entirety of needs of the user interface so we shall style as best we can. We must make sure that the styling we use for the application stands out from the other numerous app store websites and application, but must maintain a degree of simplicity in order to ensure user comfort and ease.

The interface will be engineered to be simple, clean, and sleek. It will target a wide variety of users that will be captivated by what the application provides, and it will also maintain simplicity for the user to not get lost trying to navigate through all of the functions. Obviously we cannot get a full idea of what users will like right away, but several prototypes will have to be made and will have to implement user test cases and experience trials. Hopefully at the end of the design process and by the time of deployment, all of the user cases will be considered and the interface will account for all issues and we will accommodate any user.

**6 - System Features (Functional Requirements)**

**6.1 - System will allow users to register**

|  |  |
| --- | --- |
| 6.1.1 | System shall allow a user to register |
| 6.1.2 | System shall notify user if email account already exists |
| 6.1.3 | System shall notify if the fields are invalid |
| 6.1.4 | System shall notify if the registration is done |

**6.2 - System will allow users to login**

|  |  |
| --- | --- |
| 6.2.1 | System shall allow a user to login |
| 6.2.2 | System shall allow a user to stay logged in |
| 6.2.3 | System shall notify user if the fields are incorrect |

**6.3 - System will allow users to search apps**

|  |  |
| --- | --- |
| 6.3.1 | System shall return search results of 10 |
| 6.3.2 | System shall return search results of 20 |
| 6.3.3 | System shall return search results of 30 |
| 6.3.4 | System shall return nothing if no fields exist |
| 6.3.5 | System shall return an image for the corresponding app in the results |

**6.4 - System will allow the download of apps**

|  |  |
| --- | --- |
| 6.4.1 | System shall allow users to search for a specific app for download |
| 6.4.2 | System shall allow users to download the app onto their own their. |

**6.5 - System will allow the upload of apps**

|  |  |
| --- | --- |
| 6.5.1 | System shall allow the user to upload apps |
| 6.5.2 | System shall allow users to set icon for their app |
| 6.5.3 | System shall allow users to set the date of availability |
| 6.5.4 | System shall inspect the app for errors |

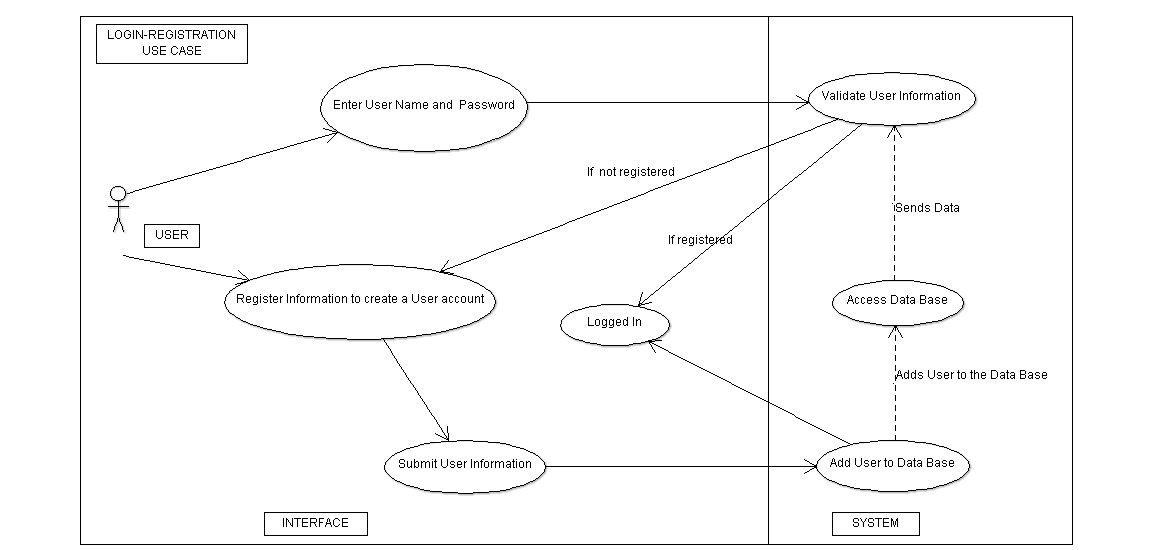
**6.6 - System will provide users with FAQs and Help Sections**

|  |  |
| --- | --- |
| 6.6.1 | System shall provide a page for users seeking frequently asked questions. |
| 6.6.2 | System shall provide a page for users to seek contact or help for the website. |

**7 - MODELS**

**7.a USE CASES**

As a user I want to register and login to the website as a user of the application.



**7.a.1 Login-Registration**

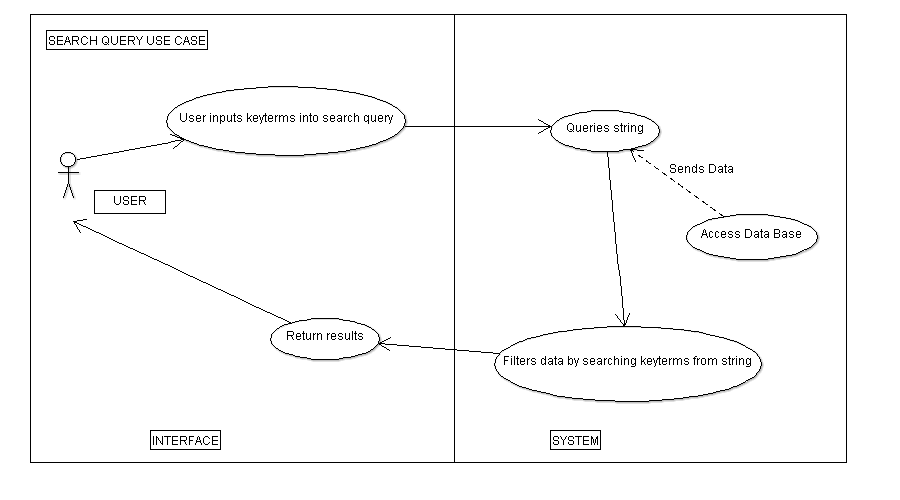
**User**: Navigate the Login and Registration page.

**System**: Displays forms with fields for login and registration.

**User**: User inputs information into Login field and Registration field. If the information inputted in the login is correct, then the user is logged in. Otherwise, they are required to register.

**System**: Validates information from the user via Database. Validates email address and checks to see if the email already exists inside the database.

As a user I want to search for my favorite or popular applications.



**7.a.2 Search-Query**

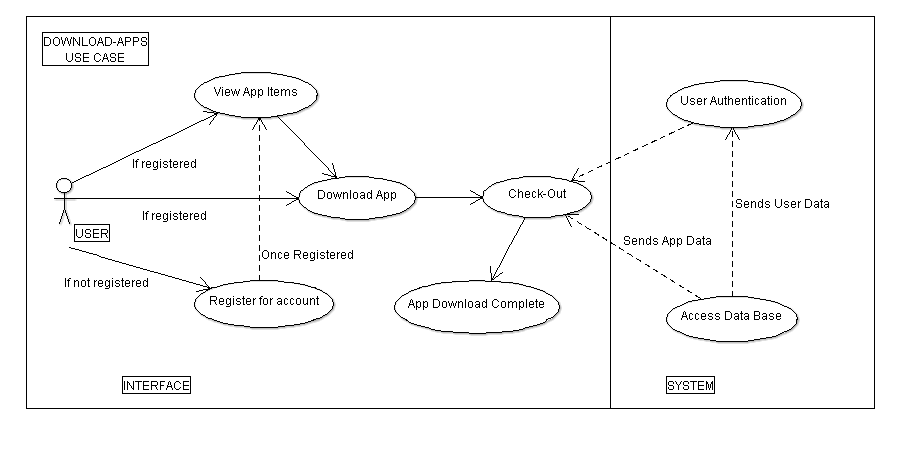
**User**: Navigates the front page of the web application.

**System**: Displays forms with fields for search query.

**User**: Inputs key terms in search bar in order to find a specific result.

**System**: System accesses database in order to find and filter results for the User.

As a user, I want to download/buy my applications.



**7.a.3 Download-Apps**

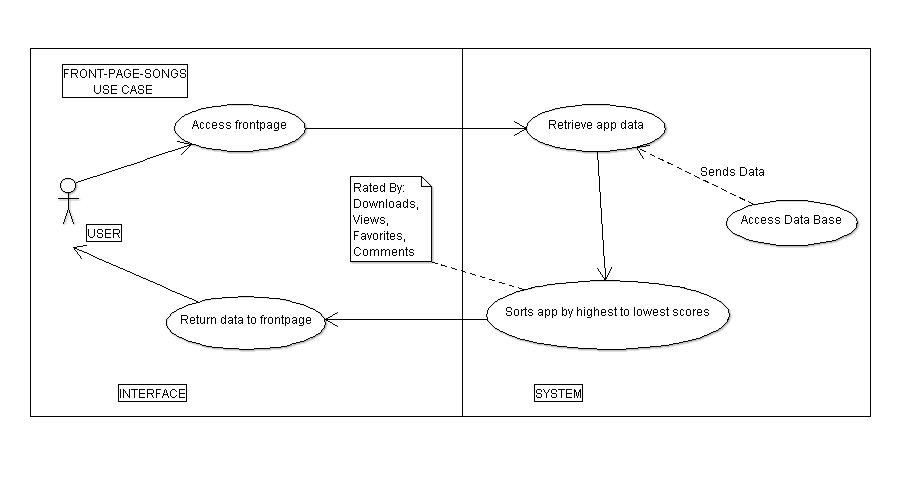
**User**: Navigates the web application inventory page. If the User is not registered for the website, then they are not allowed to make any downloads.

**System**: Displays the search query and list of apps results from the system.

**User:** User can find an app to download.

**System:** The checkout information is verified by the system. The system checks to see if the current user truly exists in the database. Afterwards, the app data is sent, the user’s information is updated, and the transaction is complete.

As a user I want see what popular applications are on the front page.



**7.a.4 Front-Page-Applications**

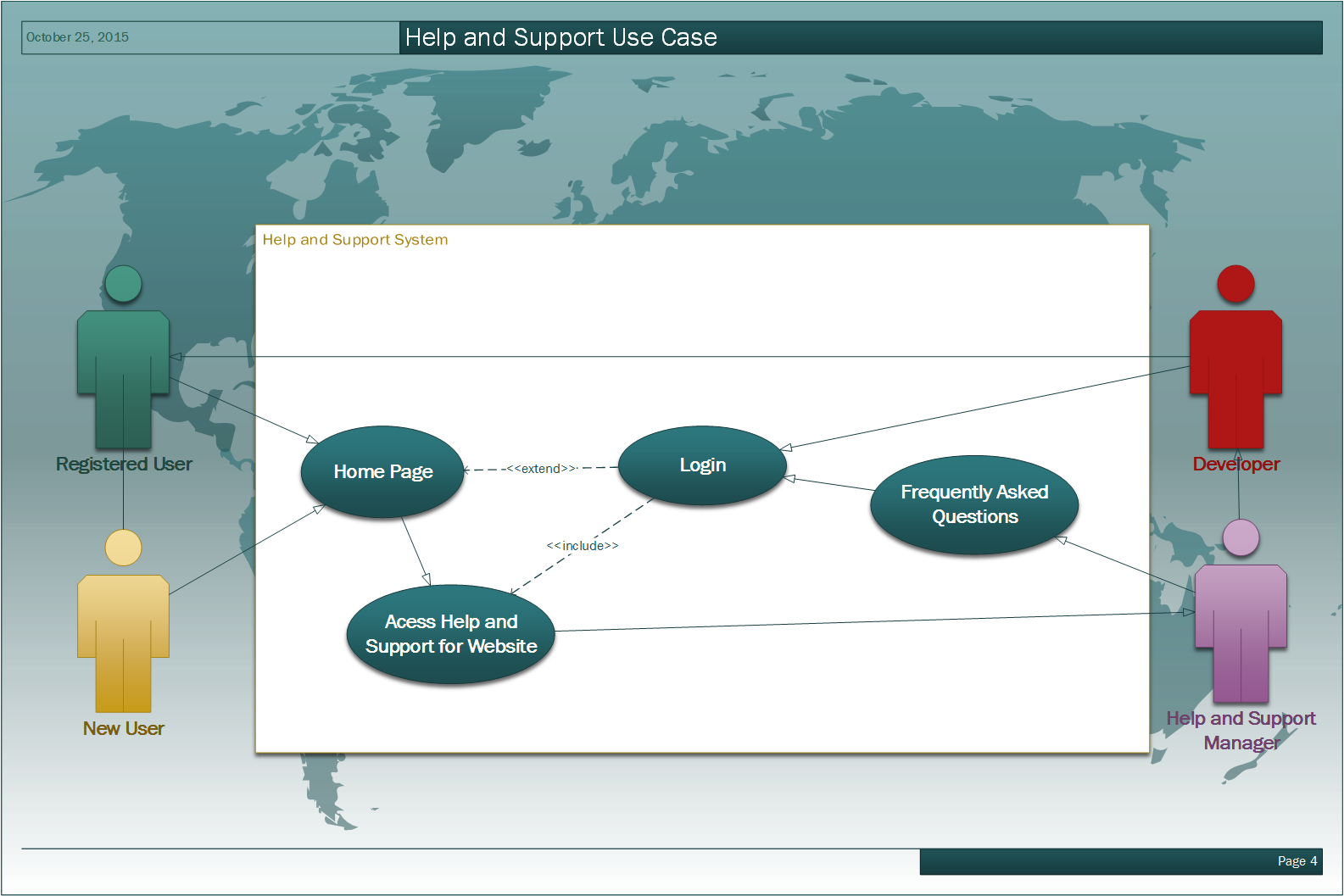
**User**: Navigates front page of the web application.

**System**: Displays the most popular apps on the front page.

**User:** Can choose which type of view such as downloads, views, popularity, favorites, comments or genre to show on their front page.

**System:** Can show different types of categories to display on the front page instead of popularity.

As a user, i want to know where to go for help and support.

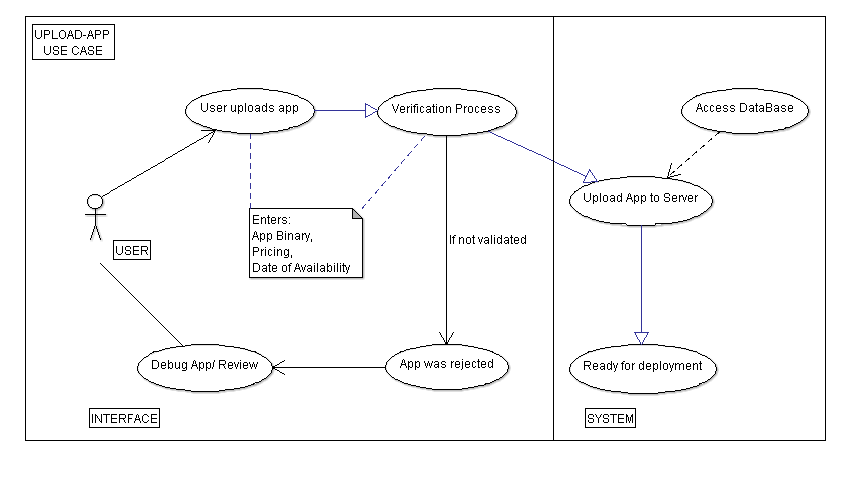


**7.a.6 Help/Support Use Case**

**User:** Finds help and support regarding information about the site such as the application’s mission, the developers of the application, the application’s about me, support contact information, and FAQ quick guide.

**System:** Will display information and links to frequently asked questions such as “How do I register for an Account?”, “Can I make a download without an account?”, and others such as “What are the requirements for an account?” Will also include contact information for any personal user related discussions.

As a developer, I want to know how to upload my apps to the website.

 **7.a.7 Upload App Use Case**

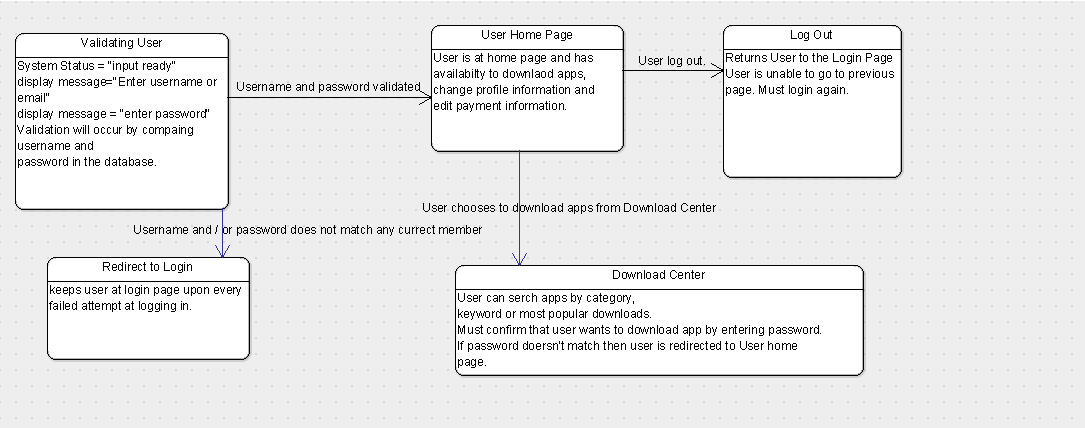
**User:** They will be able to upload their own applications to the website.

**System:** Will verify whether or not the data and information submitted is sufficient for upload.

**User:** User can debug and review their code for submission if it fails.

**System:** System will upload the application to the server for deployment if the verification passes.

**7.b STATE DIAGRAMS**

**7.b.1 Login and Index Page**

The State diagram is shown above demonstrating the layers of security used by validating the user several times. The security covered in the diagram is just the security measures on the client side.

For the login page, we of course have the user needing to provide his/her username and password. This information will be entered in the form and be used to query the database. The password and username entered must belong to a unique user in the members table of the database. If the verification confirms that the information entered is correct, then the user will be redirected to his/her home page. If the information entered is not that of any current user then the user will be redirected to the same login page. A notification saying they entered the wrong information for either the username or password will appear and a link to the registration page will show up in the case that they are a new user.

If the user is a valid user then they will be at their home page once they have logged in. At the home page the user will have the ability to search for and download apps at the download center. At the download center, the user will be able to enter and/or edit payment information for apps.

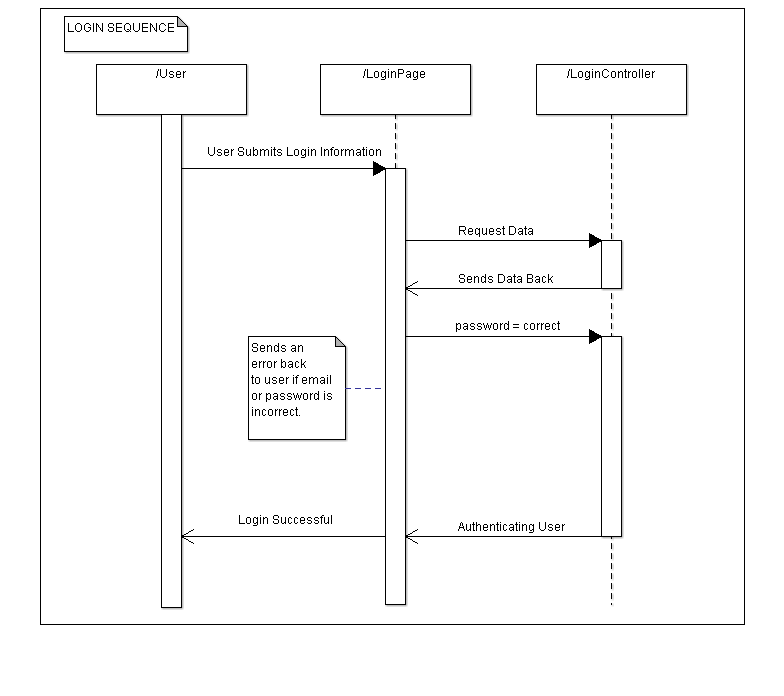
The security measures that were used on the login page will also be used when the user wants to go from the homepage to the download center. For the user to go from the home page to the download center, they must enter their username and password again to ensure that the user that logged wished to download apps. This is to ensure that whoever is on the device is still the user who logged in. Once again the username and password provided is used to query the database to make sure it is of a valid user and matches the username and password that was entered at the login page. If the username and password matches then the user is redirected to the download center. If the information entered is not that of a current user then the user remains at the home page and must try again.

Once at the download center the user will be able to search for apps by keyword, most popular or by category. Some apps will not be free and the user will need to provide payment information. The user will need to click on a button labeled "Payment information". In order to enter or edit payment information, the user must once again enter his/her username and password and that will again be queried to the database to see if the information provided is that of a current user and the user that logged in. Once that information is verified then the user can edit his/her payment information and then exit and continue to navigate through the download center.

When a user chooses an app they want to download, they can click add to the download page. Once the user clicks the confirm download button, a pop-up will appear asking for the user’s password one more time. This is to ensure that the same user who logged into the account and the download center is the same person on the device who wishes to purchase the app. Once the password is validated and it is in fact the correct password corresponding to the user then the app will commence downloading. The username and password validation is taken very seriously throughout the app in order to protect the user in the event that an unauthorized user is in possession of the device where the legitimate user logged in. There is a possibility that the user may log in and then may walk away from his/her device, giving an unauthorized user the opportunity to take advantage of being logged into the account already. However with the account verifications throughout the app, one who is unauthorized to be using the account will not be able to access payment information nor download apps without knowing the username and password despite being in possession of a device where the user already logged in. Besides the username and password validations throughout the app, there will be an activity timer that automatically logs out of an account when there has been no activity for a certain period of time. We have yet to decide how much time it should be but a reasonable amount would be 15 to 30 minutes.

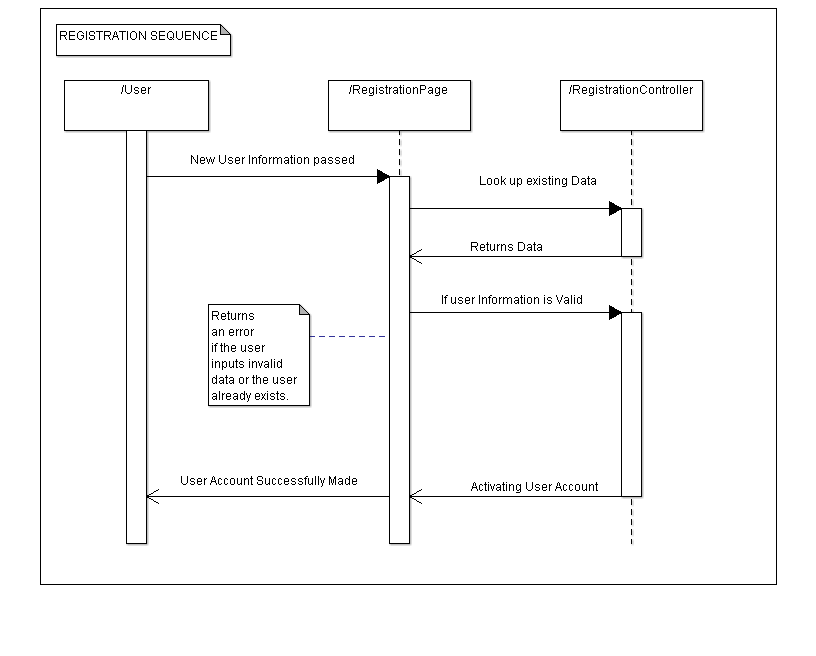
**7.c SEQUENCE DIAGRAMS**

**7.c.1 LOGIN SEQUENCE**

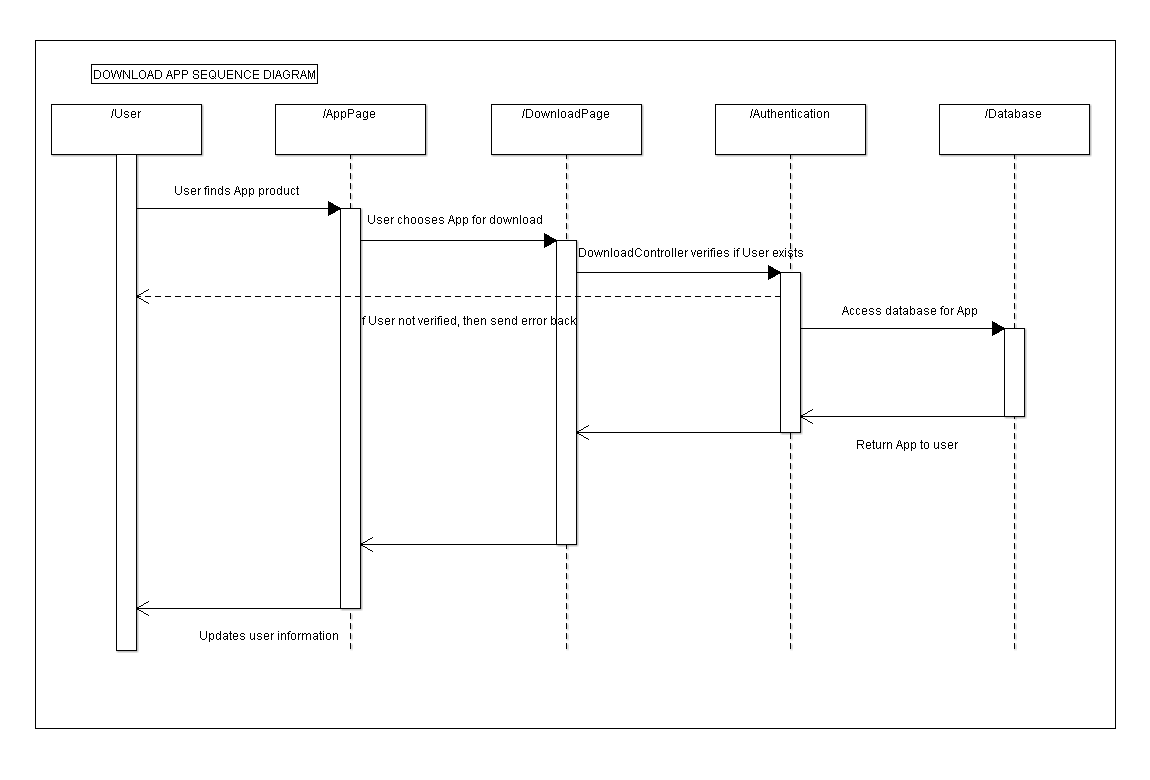


The process begins when the user pushes their email and password strings from the form into the controller. The controller then analyzes the data from the database and compares it to the user’s submission. If the email does not exist or the password is incorrect, then the user is sent back to the login screen with an error. Otherwise the controller activates the user’s account and logs them into the website.

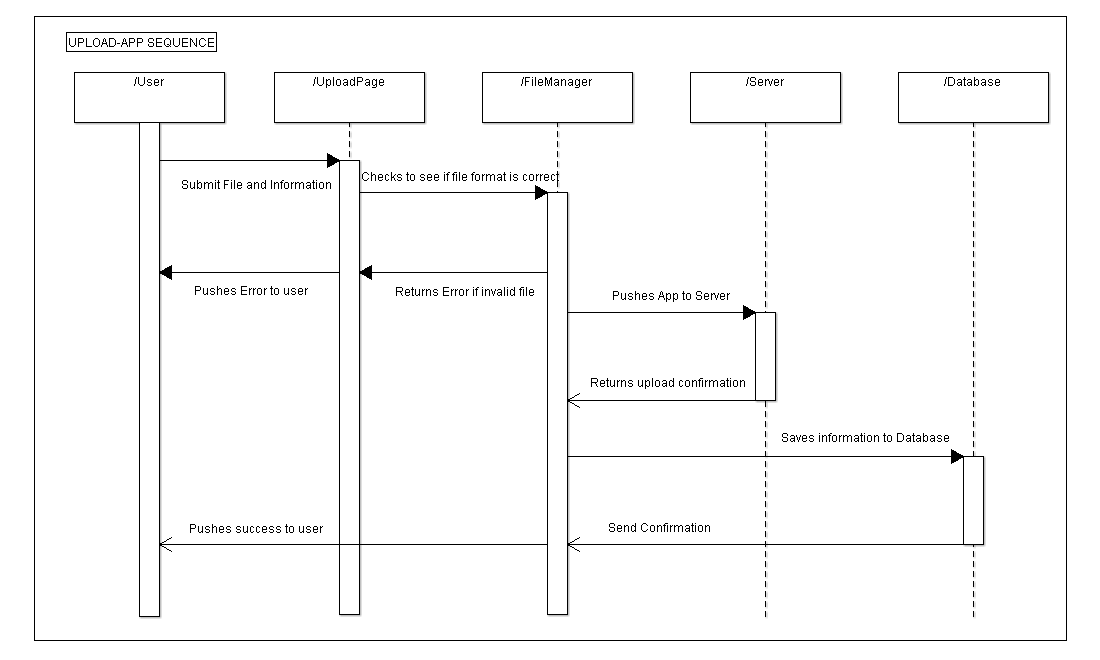
**7.b.2 Registration Sequence**



The process begins when the user pushes their information (firstname, lastname, email and password) to the website controller. The controller checks to see if the email the user sent in already exists within the database. If so, then the controller returns an error. If the data fields of other information do not satisfy the requisites, then another error is sent. Otherwise, the controller generates a new user based on the information the user submitted.

**7.b.3 Download Apps**

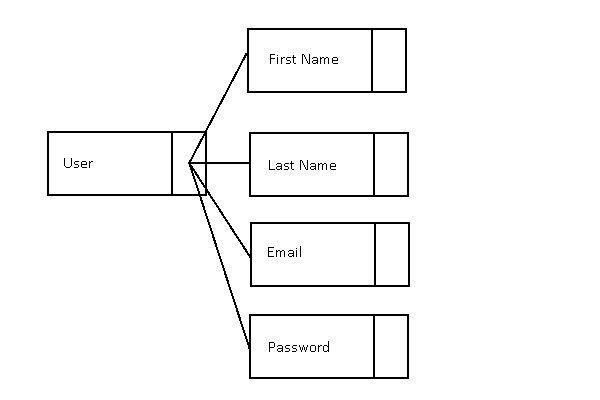
User begins thedownloading process by selecting the appropriate app for download through the search query. Once the user completes their decision, they can checkout their item. The download page and controller handles all the information of that specific app. When the user finally decides to download the app, the controller requires the user to resubmit their password for verification. If the password is incorrect, then the user is sent a error message. Else, the controller accesses the database, retrieves the app, and pushes it back to the user.

**7.b.4 Upload Apps**

Users who wish to upload apps into the website must go to the upload page. Once there, the website will ask the user to submit a file of the app, the data for publication and pricing. Once the data is submitted, a verification process runs in the controller to see if the app functions correctly. If not, a message will be pushed to the user. Else, the file is added into the server, and the database which holds all information about that app is added and will be updated if the user wishes to change anything related to it.

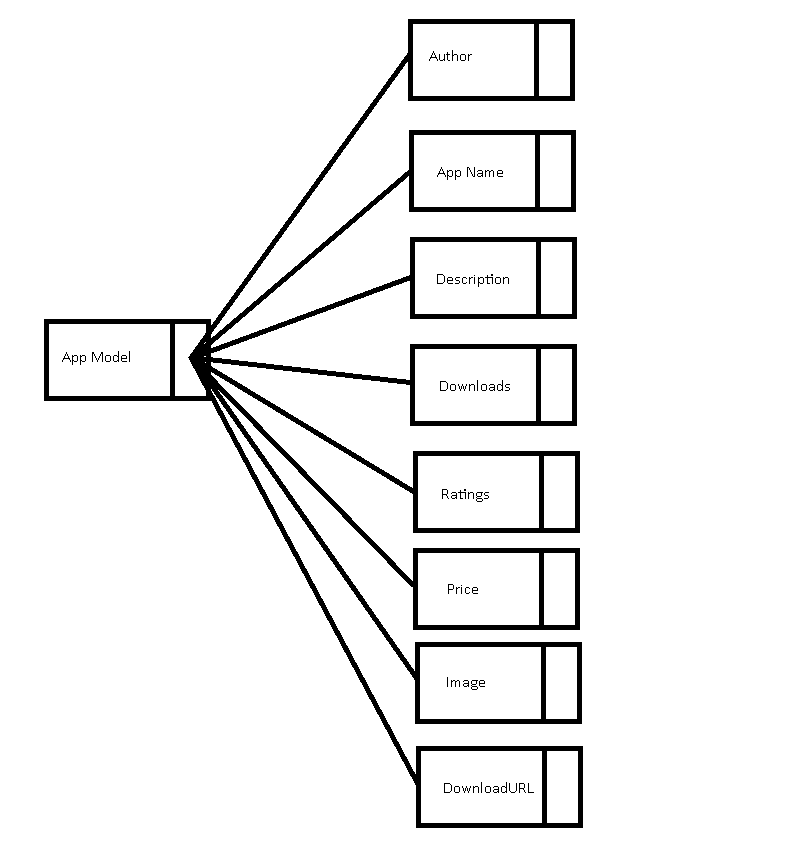
**7.d DATA TREES**

**7.d.1 User Data Tree**



Basic User Data tree when handling login and registration information. A user contains a first name, last name, email and password.

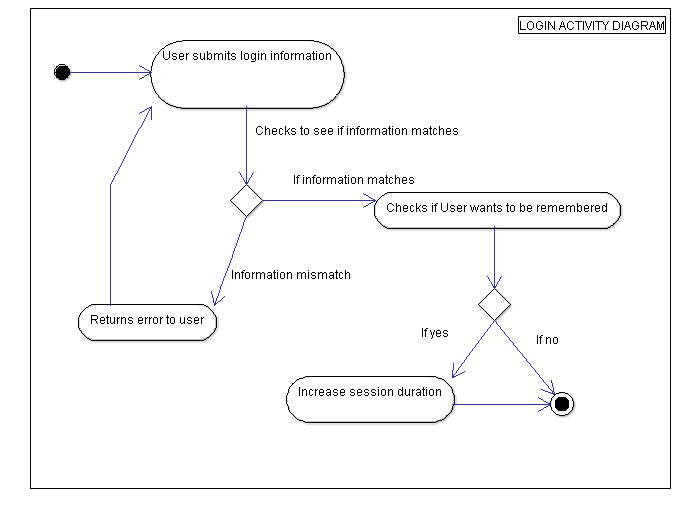
**7.d.2 App Data Tree**



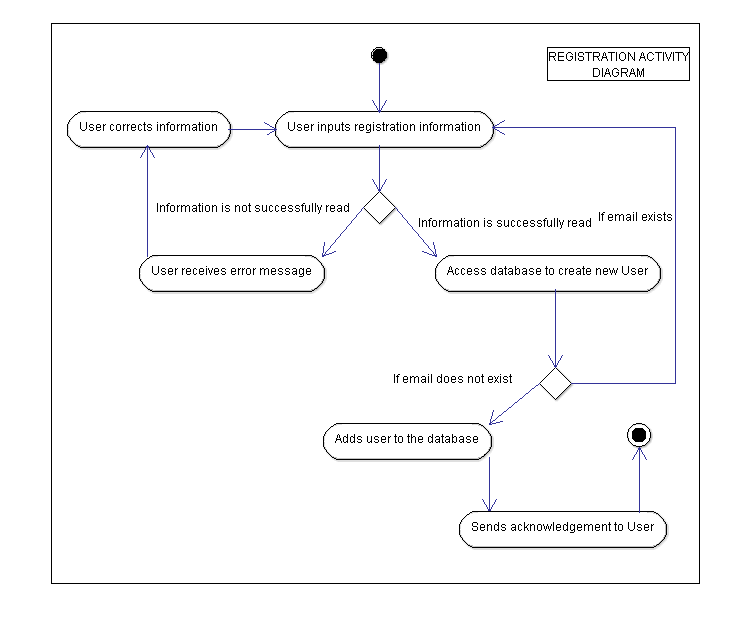
Basic App data tree for applications being sold on the website. An application class holds the file of the actual program, a description which hold information about the app and an image, and the prices for the app.

**7.e ACTIVITY DIAGRAMS**

**7.e.1 Login Activity**

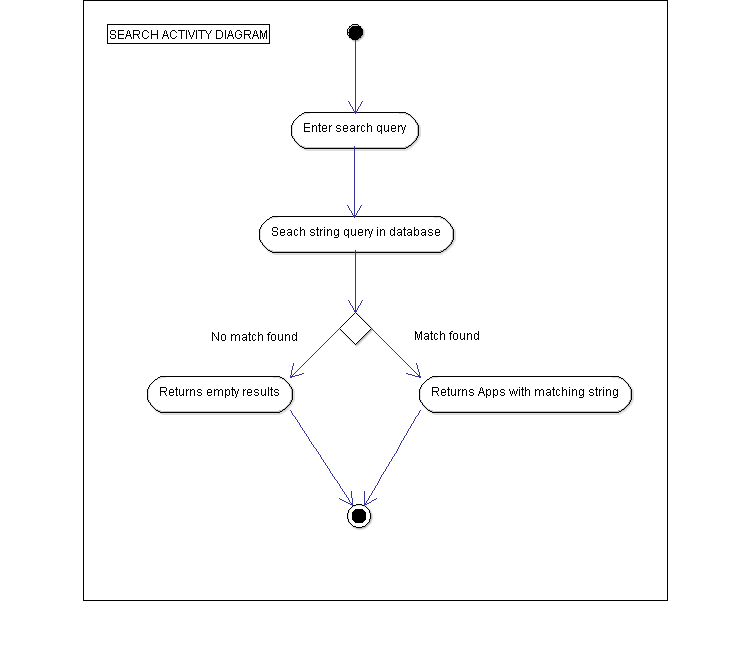


Users are able to input their email and password in order to login to the website. They are also able to check off if the website wants to remember this particular user on a particular machine. If an error is found, the controller pushes the information back to the user so they will be able to rectify their information. If the information is indeed correct, the controller will check to see if the “Remember me” box is checked. If so, then the session duration increases to one week. If not, the duration lasts for 30 minutes. The activity terminates after the user logs in.

**7.e.2 Registration Activity**

Users are able to register for the website through the registration page. Required informations are: first name, last name, email, password, and re-submitting password. If any of these fields are missing or do not meet the requirements, then an error is pushed back to the user to rectify the information. If the fields are correct, the controller access the database to see if an existing users is there. If not, the user is added to the database with the data from the registration page. Else, another error is pushed back to user to rectify the information.

**7.e.3 Search Activity**



When using the search engine, the user may enter a string query through the search box. The system will then check the database for the corresponding keys from the string and return matching apps. It will return all the apps that follow the same matching case.

**8 - Nonfunctional Requirements**

**8.1 System/Program Requirements**

Application for the app store requires a web browser of Internet Explorer 7 or above (or with similar web browsers of that capability). App store is available to all Operating systems that are able to run the required web browsers.

**8.2 Performance Requirements**

|  |
| --- |
| System is required to process user information within seconds from submission. |
| System should be able to handle files being submitted and identify which ones are acceptable or not. |
| System should be able to login users in less than 5 seconds. |

**8.3 Security Requirements**

|  |
| --- |
| System should be able to properly handle the login and logout of a user within a certain machine. |
| System is required to leave a user logged in for a maximum of 30 minutes. If the “Remember Me” is checked, then the duration increases to 1 week. |
| System shall hide text fields crucial to user’s privacy. |
| System is required to properly handle the user’s credit card information and assure that it does not appear openly in any sort of interface. |

**8.4 Purchasing Requirements**

|  |
| --- |
| System is required to respond to user request within seconds of a download request. |
| System is required to return a working version of the application or return an error if the application is corrupt. |

**8.5 Upload Requirements**

|  |
| --- |
| System is required to check to see if the app follows the required file format to be submitted. |
| System is required to update the user information for all apps he or she has uploaded. |