First Iteration and Project Plan

Advanced Software Engineering

Team: **Hunter** Team ID: 4

**Introduction**

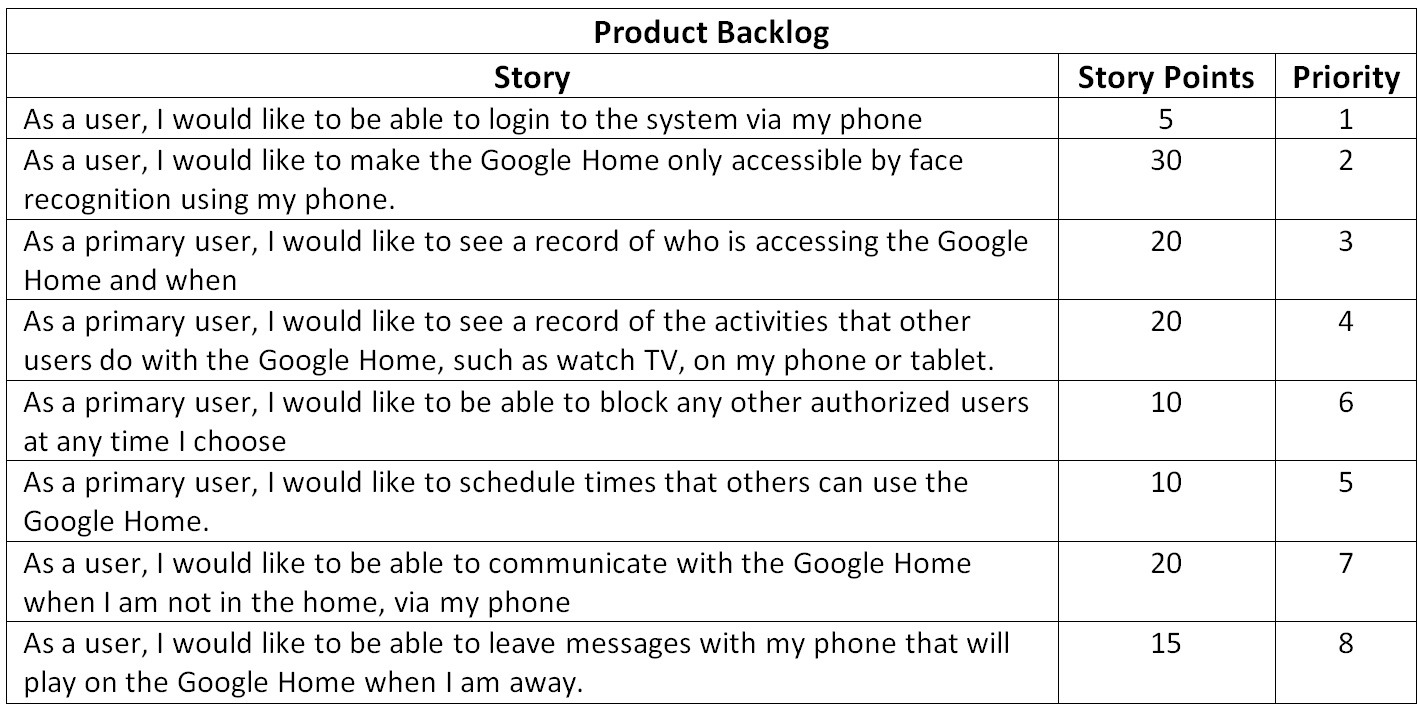
Our project involves using Google Home and the Internet of Things to build an authentication and scheduling app that can be used on an Android phone for accessing, scheduling, and viewing activity on the Google Home.

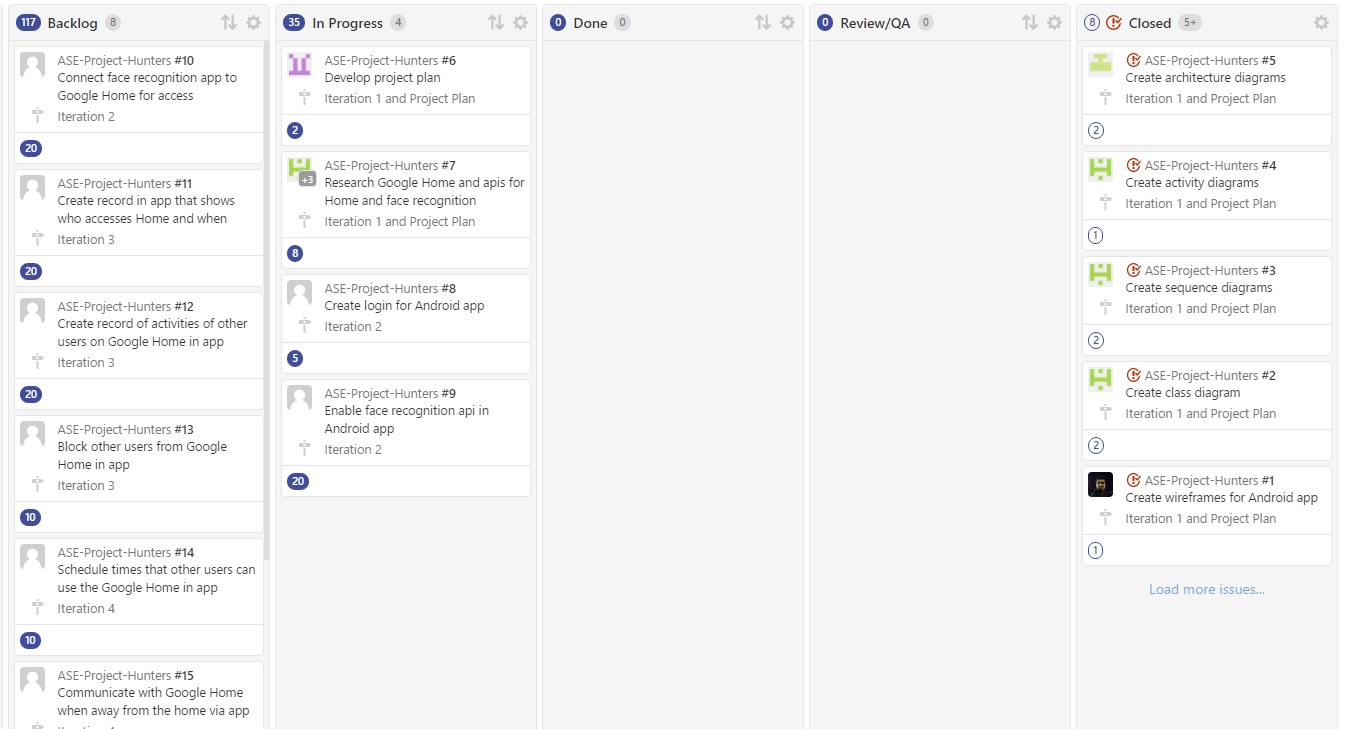
**Project Goal and Objectives (revised)**

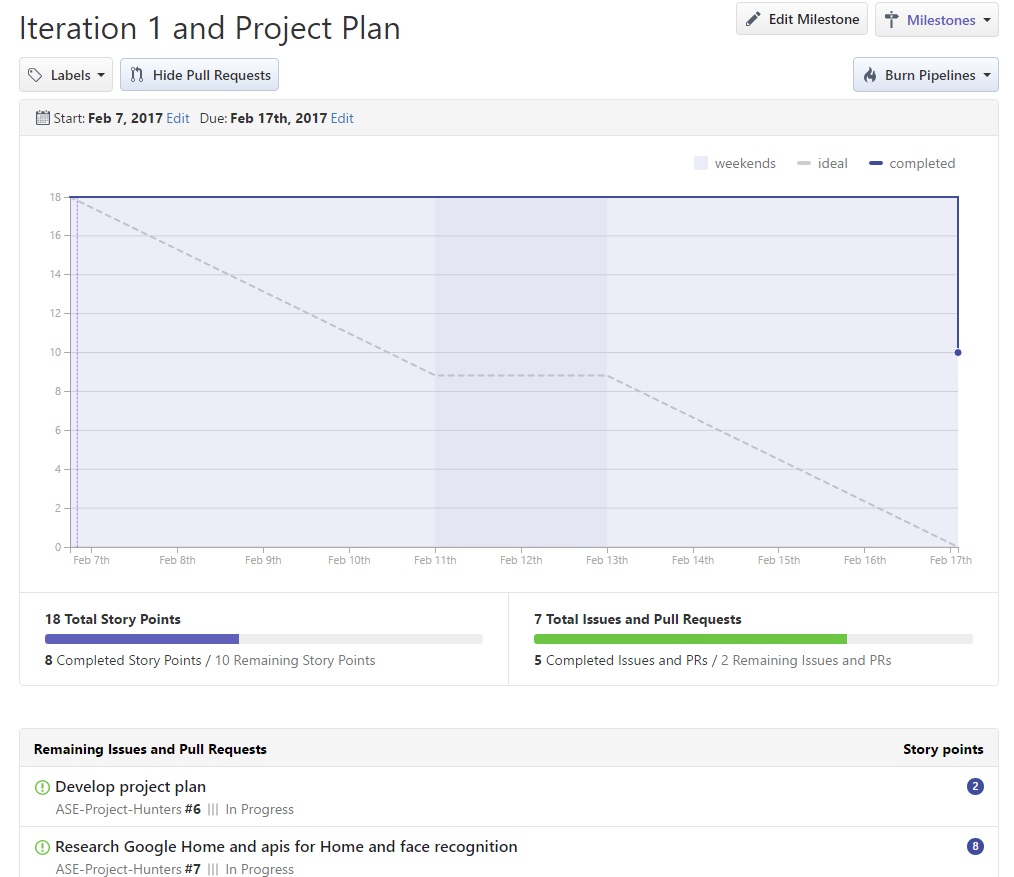
The overall objective is to build a face detection and recognition system which will be useful to authenticate Google Home. We will use a freely available api to detect faces and authorize users. There will also be scheduling and monitoring capabilities for the application.

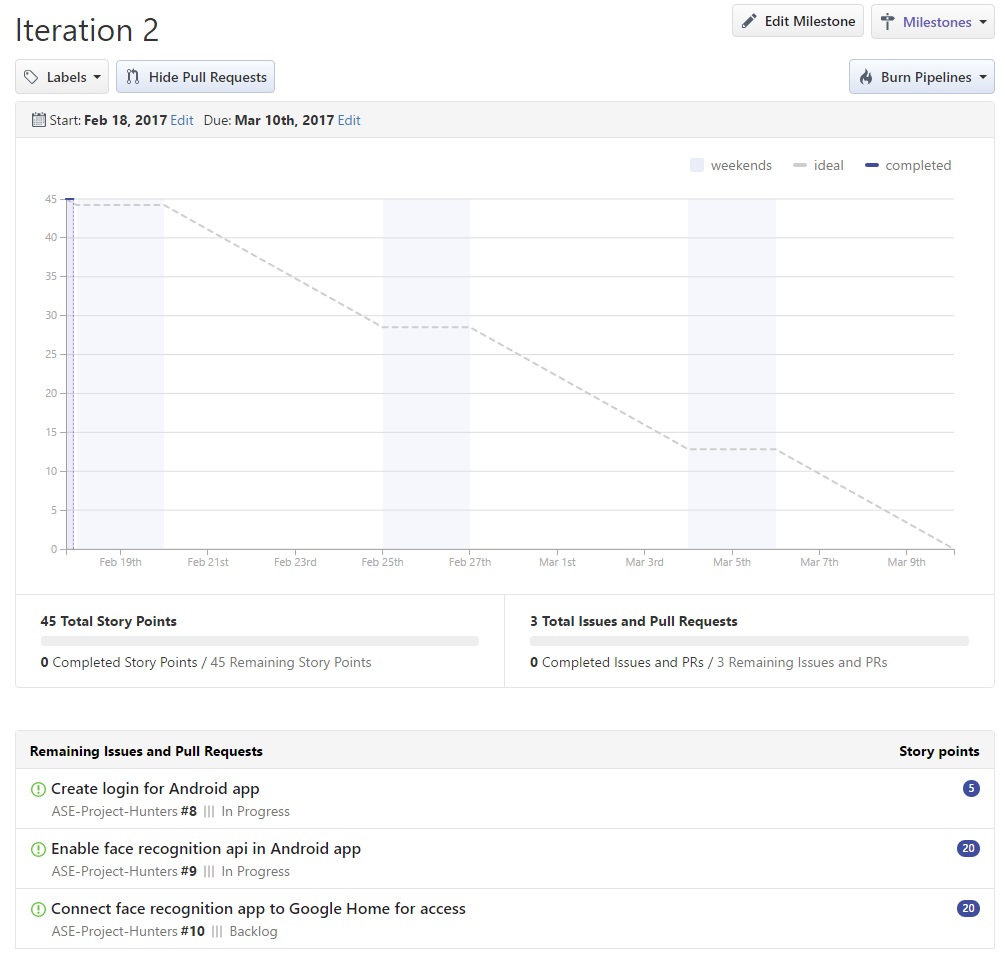
The specific objectives and features include building an Android app to detect faces and authenticate users that have been previously authorized, to view the activities of different users of the Google Home, to block or give access to different users, to schedule users or give them an allotted time to use the Google Home, and to record messages for playback on the Google Home.

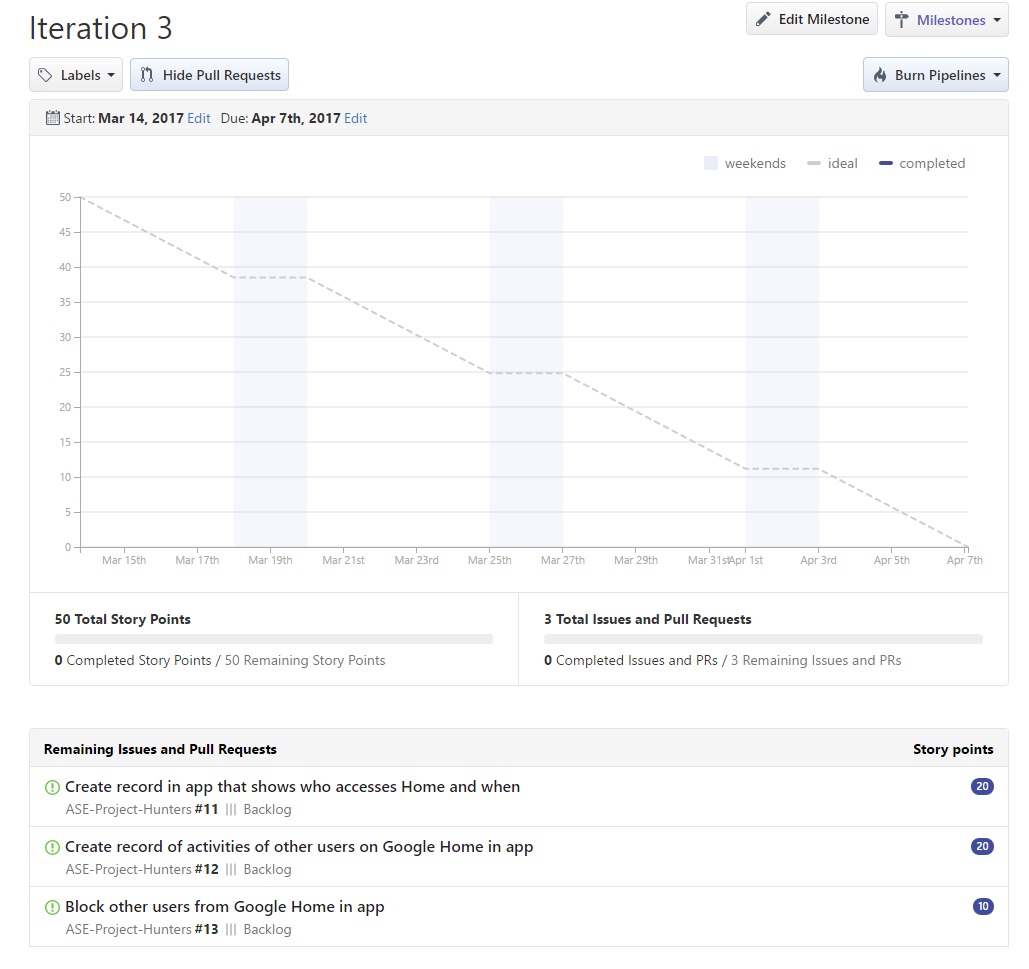
The significance of this app for Google Home is that authentication of specific users currently isn't available. During the SuperBowl this year, a Google Home commercial, saying "OK Google...", set off many Google Homes in users' houses. This app would allow only specific users to access the Google Home. It would also allow an administrator to monitor and limit usage to the Google Home, which might be helpful for parents to control their kids' TV watching, for example. Another significant use might be allowing playback of messages, so users can hear a family members' voice for reminders or emergency calls. None of these features currently exist for the Google Home.

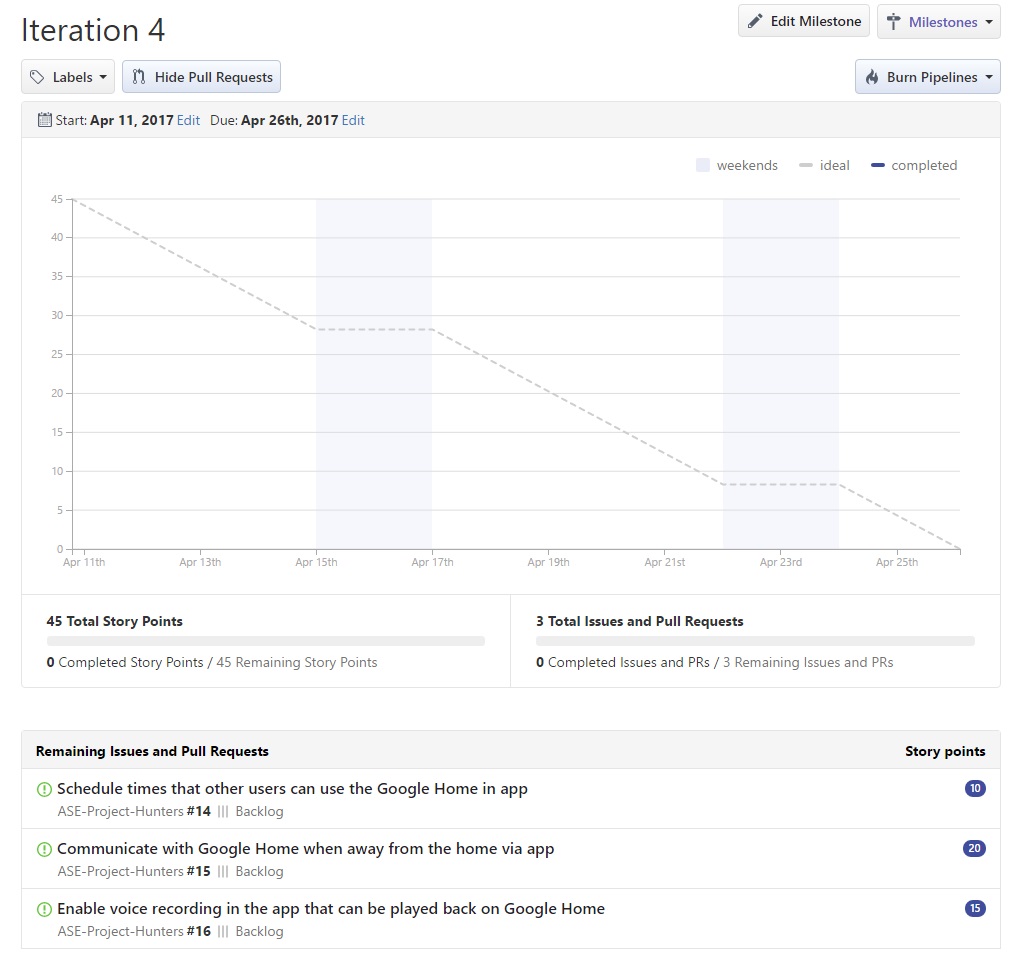
**Project Plan**   
Product Backlog  


The backlog and current issues are also visible on our ZenHub board:  
The link is [here](https://github.com/shalin51/ASE-Project-Hunters/blob/master/documentation/Iteration1-board.jpg#boards?repos=80387989)

The timelines and iteration schedules are visible in the backlog above. The team responsibilities have been detailed for the first iteration only. The burndown charts, showing the stories and time estimate for each iteration are as follows (follow the links for more detail):  
  
[Iteration 1](https://github.com/shalin51/ASE-Project-Hunters/blob/master/documentation/Iteration1-burndown.jpg#reports?report=burndown&milestoneId=2331484)  


[Iteration 2](https://github.com/shalin51/ASE-Project-Hunters/blob/master/documentation/Iteration1-burndown.jpg#reports?report=burndown&milestoneId=2331485)  


[Iteration 3](https://github.com/shalin51/ASE-Project-Hunters/blob/master/documentation/Iteration4-burndown.jpg#reports?report=burndown&milestoneId=2331486)  


[Iteration 4](https://github.com/shalin51/ASE-Project-Hunters/blob/master/documentation/Iteration4-burndown.jpg#reports?report=burndown&milestoneId=2331487)  


**Use case scenarios**  
1. Authentication and Authorization  
The user logs into the system

Actors: Google Home app user  
Precondition: the user has not yet entered the system  
Description  
1. This use case begins when a user opens up the application and is not logged in.  
2. The user is prompted to enter a user id and password.  
3. If the user's credentials are authenticated, the use case ends in success with the user reaching the application's functions.  
Alternative: an invalid user id and password are entered at step 2. An error message is displayed and step 2 is repeated until the use case succeeds to step 3.  
Open issues: how many times should they be allowed to enter invalid credentials before being denied access? How would access be reinstated? Logout when the app is closed or after a certain amount of time? Do we need a separate login if we are also using face recognition?

1. Face Recognition for Google Home authorization  
   The user takes a picture of his face in our Google Home app before he is allowed to use the Google Home features

Actors: Google Home user  
Precondition: the user has not yet been fully authenticated and wants to use Google Home  
Description  
1. This use case begins when a user has logged in and wants access to Google Home  
2. The user is prompted to take a picture of their face  
3. If the user is authenticated from a list of authorized users, then the user is given access to the Google Home  
Alternative: the face is not recognized in step 3. An error message is displayed and the user is allowed to try another picture.  
Open issues: How many pictures should they be allowed to take? Similar to login above. How would the user log out?

1. Google Home Authentication and Access Record  
   A primary user wants to see a record of who accesses the Google Home and when

Actors: a higher level Google Home user (like admin)  
Precondition: the user has logged in and been authenticated as the administrative user  
Description  
1. This use case begins when the admin has logged in and been authenticated as administrative user  
2. The user then selects an option inside the app to view a list of who has accessed the Google Home and when  
3. The use case ends when the user exits this screen

1. Google Home Activity Record  
   A primary user wants to see a record of the activities performed with the Google Home, such as who was watching TV and when.

Actors: a higher level Google Home user (like admin)  
Precondition: the user has logged in and been authenticated as the administrative user  
Description  
1. This use case begins when the admin has logged in and been authenticated as administrative user.  
2. The user then selects an option inside the app to view a list of who has accessed the Google Home and when  
3. The use case ends when the user exits the screen  
Open issues: Should this be combined with use case 3 and visible in the same menu?

1. Block Google Home Access  
   A primary user wants to block other authorized user, either temporarily or permanently, from using the Google Home

Actors: a higher level Google Home user (like admin)  
Precondition: the user has logged in and been authenticated as the administrative user  
Description  
1. This use case begins when the admin has logged in and been authenticated as administrative user.  
2. The user selects an option inside the app to view a list of authenticated users  
3. The user selects another user that they want to ban  
4. The user then selects a permanent ban or a temporary ban (e.g. a child is grounded and cannot turn on the TV without the Google Home)  
5. The use case ends when the user exits the screen  
Open issues: the admin level user wants to add authorized users. How to add their pictures? Adding users should be another use case.

1. Schedule Access Times for Google Home  
   A primary user wants to schedule times that Google Home is accessible; e.g. a child can only use it to watch TV or play music during certain times of the day

Actors: a higher level Google Home user (like admin)  
Precondition: the user has logged in and been authenticated as the administrative user  
Description  
1. The use case begins when the admin has logged in and been authenticated as administrative user  
2. The user selects an option inside the app to schedule times for other users to use the Google Home  
3. The user is presented with a scheduler tool to schedule users for different times, either on recurring or single use basis  
4. The use case ends when the user exits this screen

1. Record Messages to Play Back on Google Home When Away  
   A user wants to be able to speak over the Google Home using the phone when they are away. An example use would be that the family has their phones off. So someone away can record a message on their phone. The Google Home would make a loud beep and then play the message for the family to hear.

Actors: a Google Home user  
Precondition: the user has logged in and been authenticated  
Description  
1. The use case begins when the user has logged in and been authenticated  
2. The user selects an option inside the app to record a message to play back on the Home.  
3. The user selects a time to play the message (or to play immediately).  
4. The user can also select an option to replay the message at a certain time or interval.  
5. The use case ends when the user exits this screen.

**First Iteration Report**

1. REST API

Following is the list of apis which will be used.

* <https://westus.api.cognitive.microsoft.com/face/v1.0/detect?returnFaceId=true&returnFaceLandmarks=false>
* <https://westus.api.cognitive.microsoft.com/face/v1.0/findsimilars>
* <https://westus.api.cognitive.microsoft.com/face/v1.0/group>
* <https://westus.api.cognitive.microsoft.com/face/v1.0/identify>
* <https://westus.api.cognitive.microsoft.com/face/v1.0/verify>
* com.ibm.watson.developer\_cloud:alchemy:3.5.3

1. Detailed Design of features

Wireframes

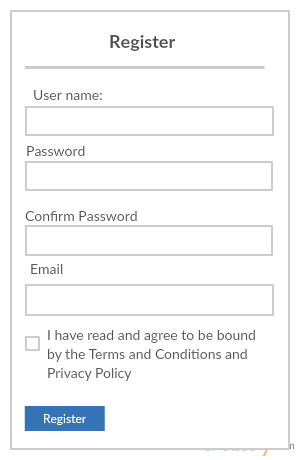


Figure 1:Register Activity

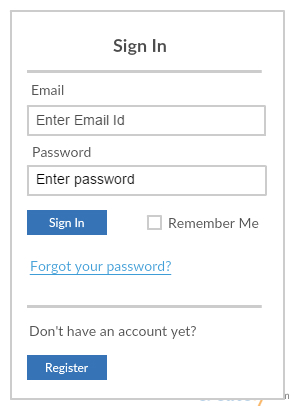


Figure 2: Sign In Activity:

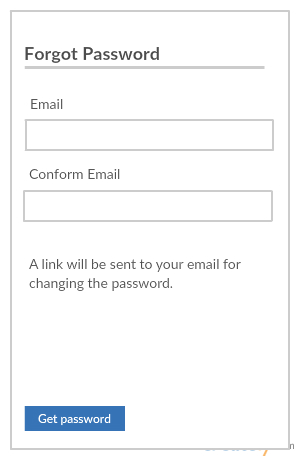


Figure 3: Forgot Password



Figure 4: Sentiment Analysis

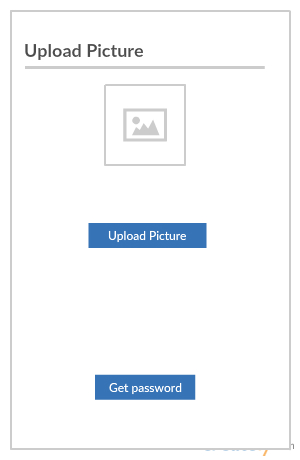


Figure 5: Upload Images

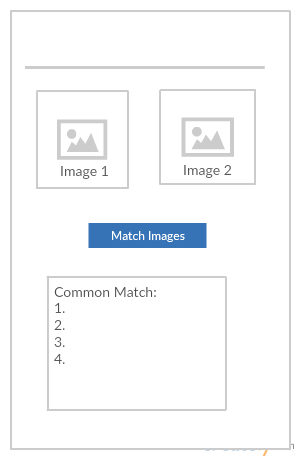


Figure 6: Compare Images

  3. Architecture Diagrams

Architecture Diagram

The user uses mobile phone and its camera to take pictures. When user wants to do any activity like comparison of image, getting face attributes, the application will pass the data to web API. Then as per retrieved result, there will be application logic which manipulates the response and displays results. For google home authentication, is the user’s validation response is true, then it will allow google home to operate. There is the provision of FaceDB which stores all images.

Figure 7: Architecture Diagram

  Activity Diagrams

In below activity diagram, if the user is very new to application, the user need to do registration. After that the user can login. The user performs any activity, like face detection, face recognition or getting face attributes. After that the given images will be compared with image dataset and validation will be done.

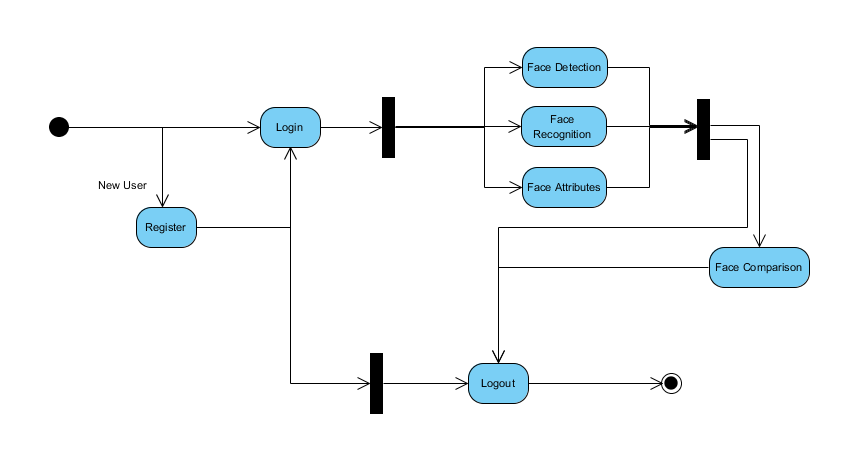


Figure 8: Activity Diagram

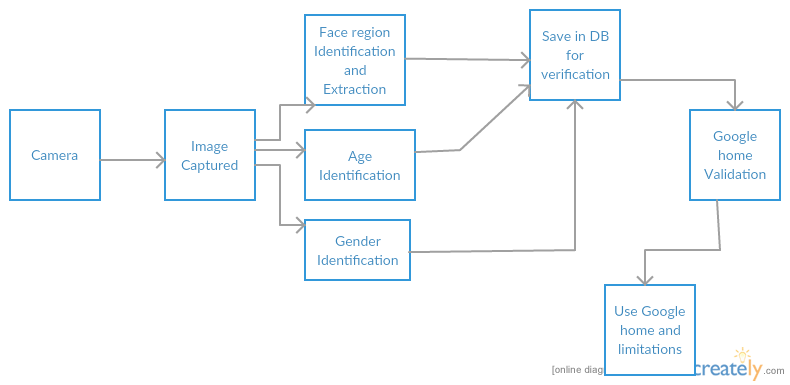


Figure 9: Activity Diagram for Google Home Validation

Class Diagram

Below is the class diagram, for the application.

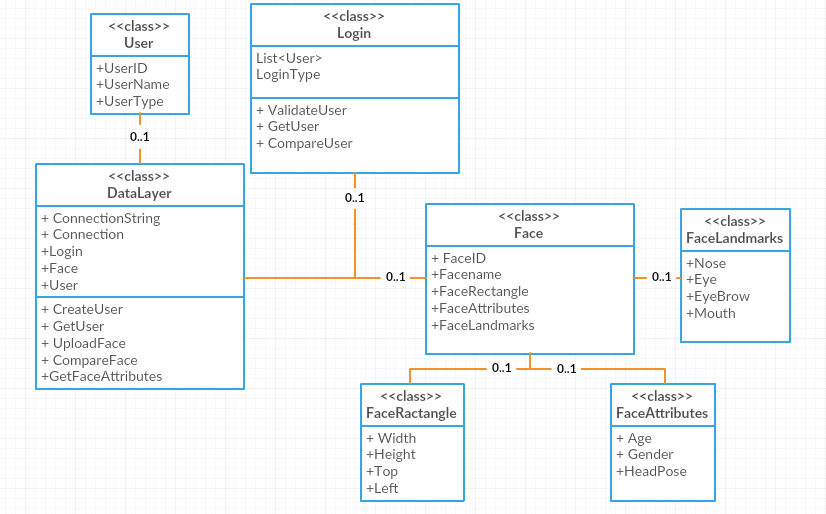


Figure 10: Class Diagram

  Sequence Diagram

For following sequence diagram, the user login. If the user is new, then CreateUser will be called from data layer and it registers the user.

For existing user, user details will be collected and passed to Validation. It fetches the user info from the database and compares. After successful login user navigates through application.

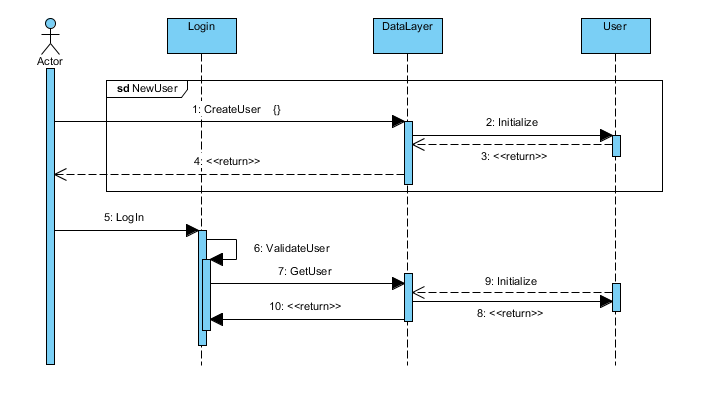


Figure 11: Sequence Diagram for login activity

1. Implementation In this project, we are going to use android.

As of now, Login activity has been done.

Below is the link where you can find source code for application.

<https://github.com/shalin51/ASE-Project-Hunters/tree/master/Source>

**5. Project Management**  
For iteration 1, we completed the tasks we set out to complete. This iteration was largely a research and planning iteration. We researched Google Home and decided what app to build for it, a face recognition and scheduling app. We completed the following tasks:  
1. Research Google Home and face recognition apis  
2. Create wireframes for the app  
3. Create class diagram  
4. Create activity diagram  
5. Create sequence diagram  
6. Create architecture diagram  
7. Develop project plan

There is nothing we didn't complete that we expected to for iteration 1. For iteration 2, we will work on stories 1 and 2 mainly, which involve the following tasks:  
1. Create login for Android app  
2. Enable face recognition api in Android app  
3. Connect face recognition app to Google Home for access

  Bibliography

**Websites**

* [https://www.google.com](https://www.google.com/)
* [https://images.google.com](https://images.google.com/)
* [https://www.microsoft.com](https://www.microsoft.com/)
* <https://www.ibm.com/watson>
* <https://madeby.google.com/home>

**Books**

* Simon, Jonathan. Head First Android Development. Farnham: O'Reilly, 2011. Print
* Johansen, Christian. Test-driven JavaScript Development. Upper Saddle River: Addison-Wesley, 2011. Print.