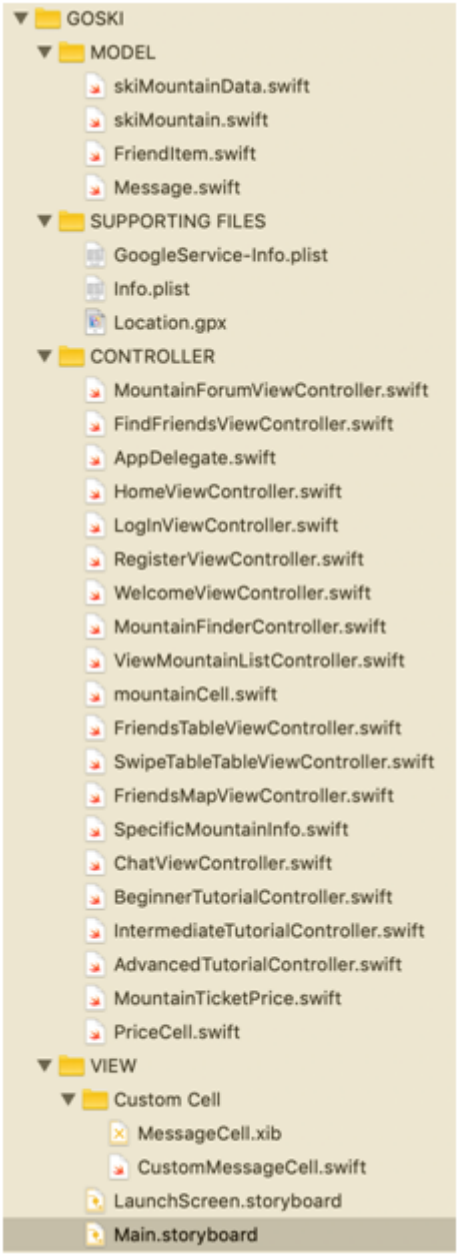
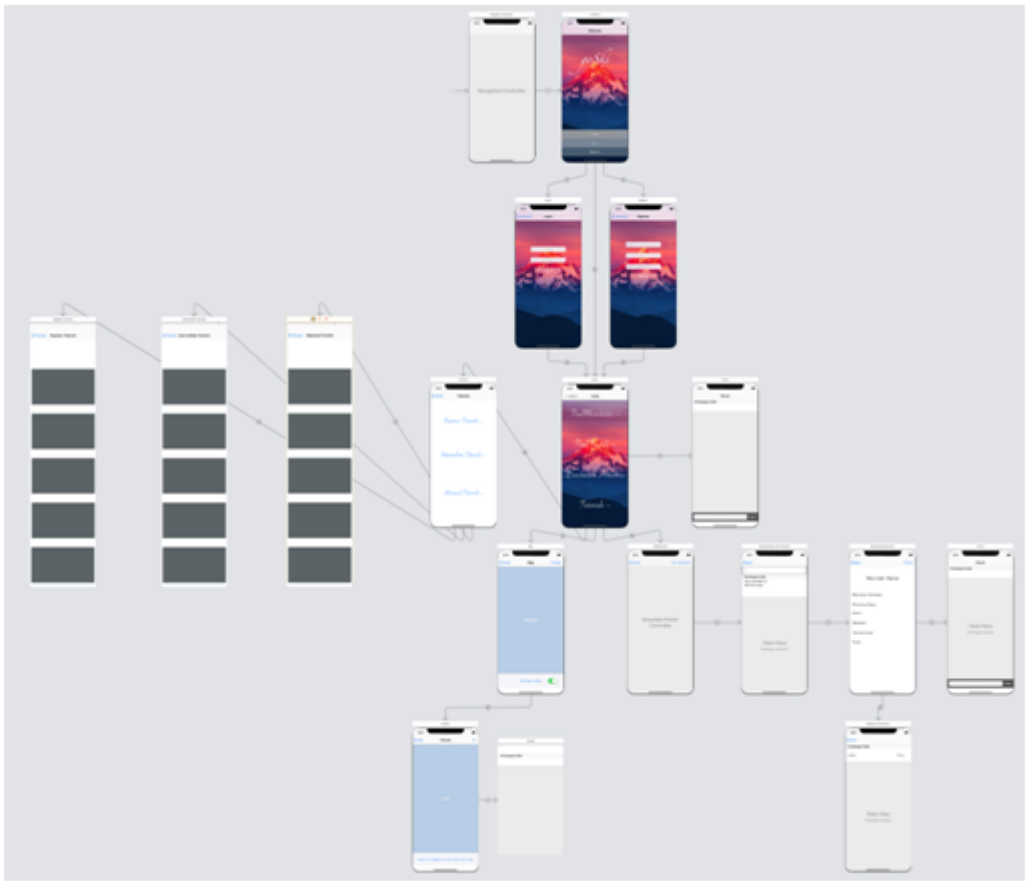
Team [0] - Beta Release

Guide for Source Code:

This part is designed for a better understanding of the relationship between classes and corresponding view controllers. We used the MVC design pattern in our project.
In here, we show classes as well as UI. We believe this part will offer tremendous help for you to understand the classes we submitted.

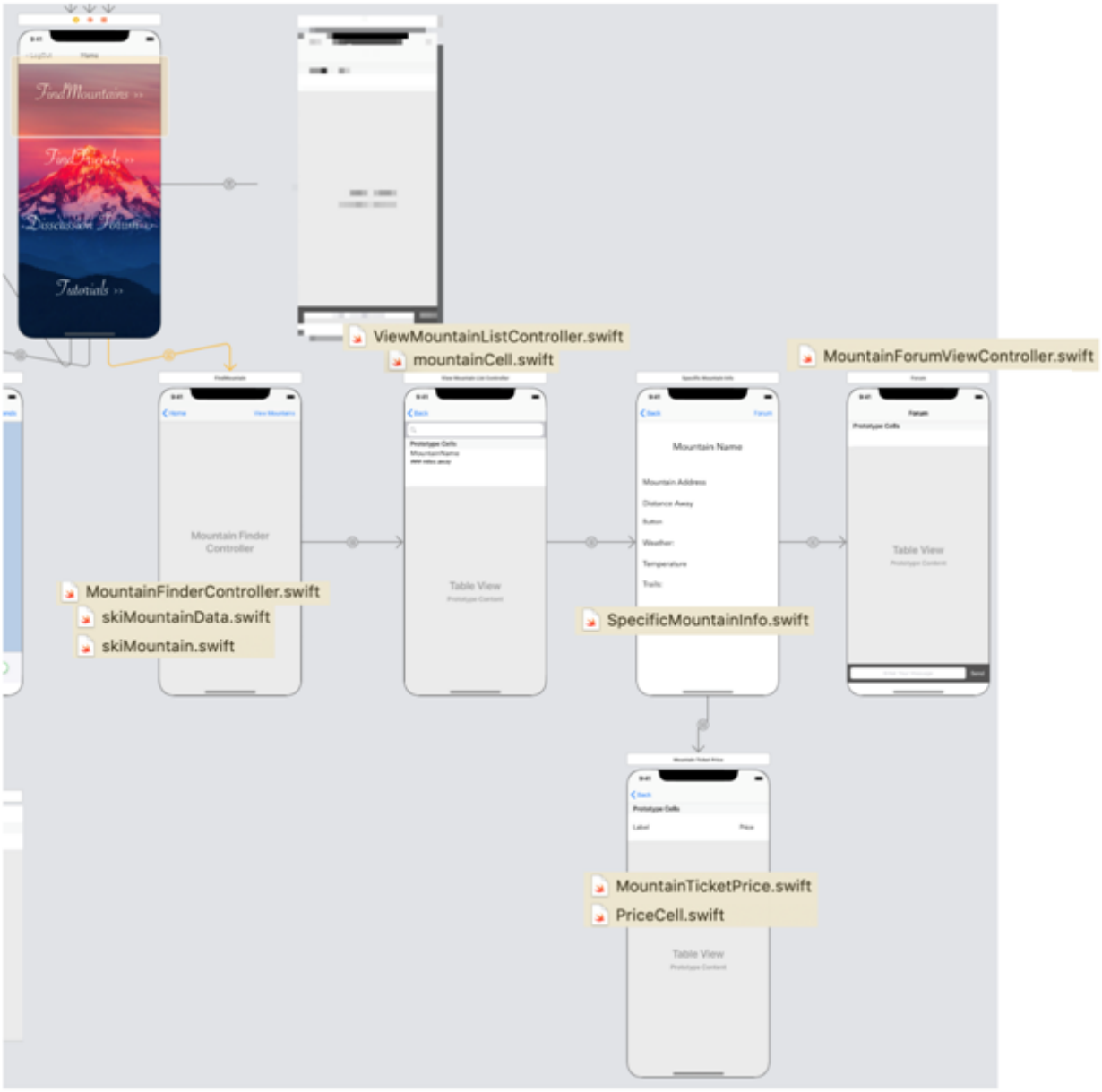




Classes for Welcome Interface including Register, Login, and Guest Login

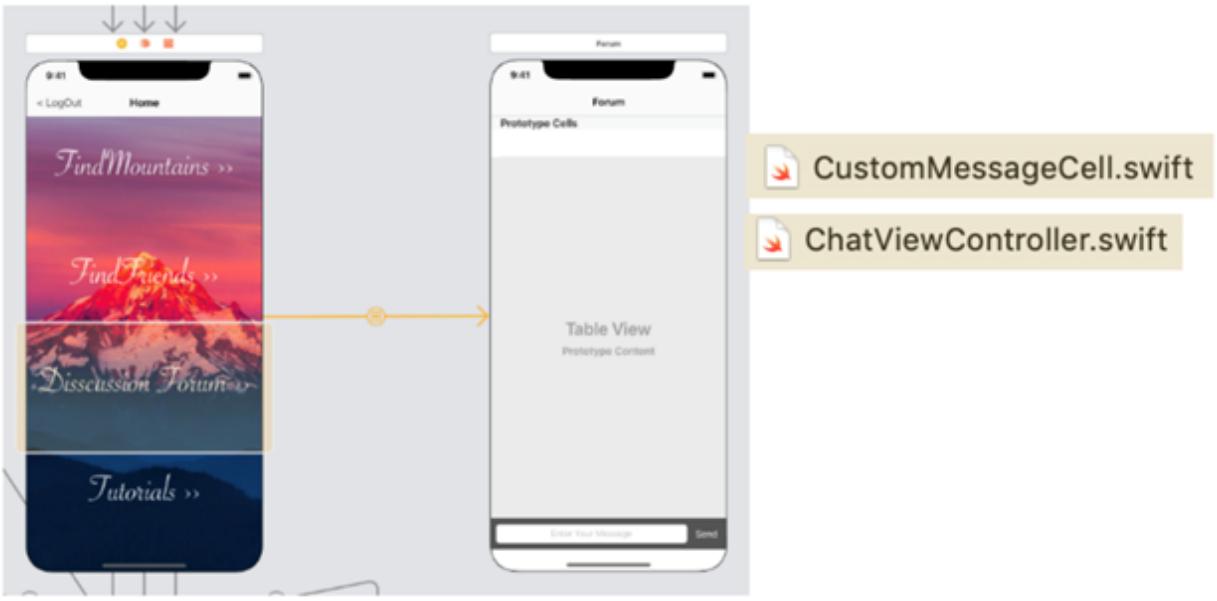
Classes for Find Friends feature:

Classes for Find Mountains feature:



Classes for Tutorials feature:

Classes for Discussion Forum feature:



Test Plan:

We have two main goals for the test:

1. Our application meets user requirements.
2. Our application works under normal circumstances.

We mainly use black box testing to test our system. We use both positive and negative testing which means we will test both normal cases and corner cases to ensure the robustness of the system. During testing, we have unit testing to test the reliability of each class and integration testing to test the whole system.

We use Xcode Simulator to simulate iOS devices. By installing our system on this simulator, we can run and test our system on different models of iOS devices. At the same time, we also install our app on real iOS devices including, iPhone X and iPhone 8. We have five people to test our app. Four of them will focus on the unit testing and one of them will focus on the Integration testing.

During testing, we will pay attention to the responses of the system to make sure they meet the user requirements. Also, we will document bugs or errors and fix them to meet the second goal.

Schedule:

Date	Tester	Task	Test Cases	Methods
04/16/2019	Haoran	Test the Login, Register and Guest functionality.	TC1-5	Black Box Testing Unit Testing
04/17/2019	Eric	Test "Display Mountains" component Test "Mountain Information" component	TC6-7	Black Box Testing Unit Testing
04/18/2019	Haoran	Test "FindFriends" component	TC8	Black Box Testing Unit Testing
04/19/2019	Damin	Test "Tutorials" component	TC10	Black Box Testing Unit Testing
04/20/2019	Huiming	Test "Discussion Forum" component	TC9	Black Box Testing Unit Testing
04/21/2019	Shuze	Test the whole app.	TC1-10	Black Box Testing Integration testing

04/22/2019		Have a meeting and fix bugs		
04/23/2019	Shuze	Test the whole app.	TC1-10	Black Box Testing Integration testing
04/24/2019		Further improvements		
04/25/2019		Final Release		

Test Cases

TC1: Register [Successful Registration]

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) has available space.

Sequence of Actions and **Specific Expected Results:**

1. Click Register button on the welcome interface.
   GoSki should show an interface to allow user input email and password.
2. Input an email address, passwords for twice.
   GoSki should display the email explicitly and display the password as dots.
3. Click Register button
   GoSki should register an account for the user and redirect the user to the main menu.

TC2: Register [Duplicate email]

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) has available space.

Sequence of Actions and **Specific Expected Results:**

1. Click Register button on the welcome interface.
   GoSki should show an interface to allow user input email and password.
2. Input an existed email account, the password and confirmed password.
   GoSki should display the email explicitly and display the passwords as dots.
3. Click Register button
   GoSki should display a window to tell the user this email has been registered.

TC3: Login [Successful Login]

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected

Sequence of Actions and **Specific Expected Results:**

1. Click Login button on the welcome interface.
   GoSki should show an interface to allow user input email and password.
2. Input an existed email account and the password.
   GoSki should display the email explicitly and display the passwords as dots.
3. Click Login button
   GoSki should display the main menu.

TC4: Login [Email and password do not match]

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected

Sequence of Actions and **Specific Expected Results:**

1. Click Login button on the welcome interface.
   GoSki should show an interface to allow user input email and password.
2. Input an existed email account and password.
   GoSki should display the email explicitly and display the passwords as dots.
3. Click Login button
   GoSki should show the Login Failed window to indicate the password and the account do not match.

TC5: Guest [Successful Guest Login]

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected

Sequence of Actions and **Specific Expected Results:**

1. Click Guest button on the welcome interface.
   GoSki should show a window including tips for guests and display the main menu. The Find Friends and Discussion Forum should not be able to used by guests.

TC6: DisplayMountainsOnMap

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected.
Location function is allowed.

Sequence of Actions and **Specific Expected Results:**

1. Click FindMountains on the main menu.
   GoSki should display near mountains on a map and the user's current location.

TC7: ViewMountainsInformation

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected.

Location function is allowed.

Sequence of Actions and **Specific Expected Results:**

1. Click ViewMountains on a mountain map interface
   GoSki should display a list of near mountains.
2. Click a mountain entry.
   GoSki should display a interface of the mountain including an address, price, and weather information.

TC8: FindFriends

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected.
Location function is allowed.

Sequence of Actions and **Specific Expected Results:**

1. Click FindFriends on main menu.
   GoSki should display an interface including map.
2. Click Friends button on this interface.
   GoSki should display a list of this account's current friends.
3. Click AddFriends button.
   GoSki should show a window to allow the user to input an email address of a target friend.
4. Input the email address of target friends.
   GoSki should show a window on the target friend's interface.
5. Click Accept.
   GoSki should update the friend list to include each other in both users' friend list.
6. Click Share Location
   GoSki should display each other's location on the map.

TC9: SendMessageInDiscussionForum

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected.
Users have logged in GoSki with an account.

Sequence of Actions and **Specific Expected Results:**

1. Click Discussion Forum
   GoSki should show a forum.
2. Input a message and click "Send"
   GoSki should display this new message which can be viewed in other users' devices. This messge should include the email address and the content.

TC10: Tutorials

Preconditions: App is installed in an iOS device with 10.0.0 or higher version.
The database(Firebase) is connected.

Sequence of Actions and **Specific Expected Results:**

1. Click "Tutorials"
   GoSki should a list of videos with links categorized by skill levels.
2. Click a video.
   GoSki should redirect the user to the video website.

Test Result Template

Testing Date: 04/XX/2019 13:00
Tester: Shuze
Used Testing Methodologies: Black Box Testing, Integration testing

Test Case	Result	Comments
TC1	Pass	
TC2	Fail	Layouts are different in different devices. "XXXX" buttons cannot be clicked in XXX devices.
TC3	Fail	The system displays wrong information.
TC4	Pass	

Code Review

We were reviewed by team "Pass the Aux".

We received comments that we should add more comments to our code. So, after interim release, we have added enough high-quality comments. Now, the amount and quality of the comments are great. We have added clear comments for every important statement.

We received comments that our project has a clear organization. We followed the SOLID principles and used the MVC design pattern. For each function, we implemented a class and give this class a very understandable name. This makes our code clear and readable. Thus, our project has a good organization. Also, for each parameter, the name indicates the meaning of the parameter.

For consistency, the style of our code is well-formatted and consistent. There are three complex loops in our codes. Several names of classes or variables should be changed to meet the camel casing. We think these two suggestions are good and will make changes and improvements when we are doing further improvement.

Another comment is that in the main menu, the words do not look like button. For this comment, we add an arrow at the end of each botton to indicate that users can access this function by clicking it. We have finished this improvement.

We reviewed team "Pixel Pass".

There codes are well commented and readable. They have enough comments for their project. And the quality are very good. This makes their project easy to understand.

Since they are building a website, there not really any object oriented principles or design patterns in their code. But the overall organization is good.

Their codes are consistent. Their styles follow are stantard. The name of parameter follows camel casing.

We suggested that they can added numeric value to choose the color. They can provide some samples to give users some creative ideas and they may provide tags for buttons to show the function of the button. Also, they should make "save" and "submit" button more clear.

Contribution Summary

Stakeholders' Name	Contributions
Eric Partridge	1. Improved Mountain Information interfaces. 2. Implemented the Tutorials function. 3. Improved the readability of codes.
Shuze Liu	1. Designed test plan, test cases. Wrote the documents. 2. Improved documents of Sprint 4.
Haoran Hu	1. Implemented Forum. 2. Improved the readability of codes. 3. Improved the main menu.
Damin Xu	1. Improved the database. 2. Imported introductions of mountains.
Huiming Chen	1. Further implemented the website.

Status Report

We are following our schedule and have changed the main menu for Beta Release. We added Forum, Tutorials, and Mountain Information functions. Moreover, the structure of the database has been revised and improved. More data of mountains has been imported into Firebase. We also designed a test plan, test cases and a test schedule. We finished the corresponding documents for Beta Release and helped review other team's project. We greatly improved the readability of our codes by improving the structures of codes and adding comments. In our Github, we built a good readme file as an introduction and attached a picture and a gif of our project.