

Development Plan

Hairesthetics

Team 18
Charlotte Cheng
Marlon Liu
Senni Tan
Qiushi Xu
Hongwei Niu
Bill Song

Sept 22, 2022

Table 1: Revision History

Date	Developer(s)	Change
Sept 22	Charlotte, Marlon, Senni, Hongwei, Bill, Qiushi	Initial Draft

This document provides the timeline of the project Hairesthetics, and aims to plan the future phases of our project by scheduling team meeting plans, team communication plans, and project deliverables, as well as the steps we must take to complete each phase of work.

1 Team Meeting Plan

The team will meet twice a week for three hours at a time. The first meeting happens each Monday from 7 to 10 pm. The date and time for the second meeting will vary based on members' availability, but it will always run for at least three hours. Meetings will take place on discord virtually on the capstone server. All team members are expected to show up and join the meeting at or before the designated meeting time. If any team member is not available, the team meeting shall be rescheduled to another time. Work should be assigned to each team member properly. Team leader Marlon Liu will chair the meeting and shall have an agenda set for all meetings in advance. All decisions will be announced in the discord channel and documents will be uploaded to the capstone GitHub repository.

2 Team Communication Plan

All important Capstone related information will communicate through Discord channel, discord will also be used for virtual meetings.

If a matter is urgent, such as if a member is not at a meeting, we may choose to contact that group member via WeChat. We are choosing to use WeChat as our primary means of communication since we are all familiar with using WeChat.

3 Team Member Roles

Table 2: Team Member Roles

Name	Developer(s)	Description
Marlon Liu	AR and Backend Developer	3D simulation and facial feature detection model
Charlotte Cheng	iOS Developer	UI/UX, Ar Kit
Hongwei Niu	Backend Developer	Computer Vision
Bill Song	Backend Developer	2D algorithms and APIs
Senni Tan	iOS Developer	UI/UX, Ar Kit
Qiushi Xu	Database developer	Manage and process data

4 Workflow Plan

- The GitHub repository will be where the code and documents are kept at. The main branch should have our working version of the application. And a branch named "development" should have our latest change in the next delivery cycle. Those two branches are protected and requires pull request to be merged into.
- Every team member should branch off the development branch with the feature name when they are writing new features.
- Each team member can only make changes to the project by creating pull request and each team member should pull the latest code from the development branch before they create the pull request.
- An issue should be opened for tracking the development of features using the GitHub issue template for development tracking.
- An issue should be raised in case of errors using the GitHub issue template for error to ensure that useful information is provided for the developers to debug. Logs and track back should be provided as well.
- Issues should be tagged properly (cycle number, bugs or features, estimated resolving time, priority level)

5 Proof of Concept Demonstration Plan

5.1 Most significant Risk

The most significant risk of the project will be having the application crash on the user's mobile, for example, the application suddenly stops and go back to the mobile desktop. To overcome this issue, we have to make sure the application is tested throughout and rigorously. Another major concern is that mobiles with old iOS versions may not be supported, it is important to implement exception handling for such scenarios.

5.2 Will a part of the implementation be difficult?

Technologies such as AR, Computer Vision, and iOS development are relatively new to the team members since the team does not have sufficient background knowledge in these fields.

5.3 Will the facial feature detection be accurate?

The accuracy of the location of facial features is crucial for fitting the virtual hairstyle, since the system needs to know where those coordinates are in order to shift and rotate hairstyles in real time to match the position of the user. This might requires the accuracy of the model and the face contour algorithms.

5.4 Will testing be difficult?

Unit testing will not be difficult because each developer will be responsible for testing their individual components. There are many useful testing frameworks for the technologies used in this project, for example, pytest for Python and XCTest for Swift.

5.5 Is a required library difficult to install?

No, the required library is not difficult to install. For back end, importing a library only requires the developer to install the package and add the path to the python routine; for front end and user interface, the Swift project has a package manager to help install libraries.

5.6 Will portability be a concern?

Portability will not be a concern, since the iOS application will only be onboarded on iOS Apple App Store. Only those users have the portable mobile can access and install the application.

6 Technology

- Programming languages: Swift, Python
- Framework: Flask
- Linter tool (if appropriate): SwiftLint
- Unit testing framework: IOS unit test, IOS UI test, pytest.
- CI/CD tool: Jenkins
- Code coverage measuring tool: Coverage.py, XCode coverage checking tool.
- Hosting: AWS
- Specific performance measuring tools: XCode Organizer
- Libraries: iOS AR Kit, OpenCV, Scikit-Learn
- Version control: git
- Documentation: Javadoc, Doxygen
- Tools: XCode

7 Coding Standard

We decide to use the pep8 coding standard for our python code, Google Swift Style Guide for the Swift code. Naming convention for variables and method names is camel notation. All constant variables names should be capitalized and the values are set as type final. The code should contains a reasonable amount of comments. For example, each file should begin with last revision date and the purpose, each method should be documented with purpose, input type and output type, etc.

8 Project Scheduling

The team will have a meeting at least one week in advance of a deliverable deadline to discuss the details and work breakdown. We aim to finish and upload each deliverable one day before its deadline. Each team member will review the content of each deliverable before pushing to the main branch.