

System Design for Hairesthetics

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

January 19, 2023

1 Revision History

Date	Version	Notes
Jan 12	1.0	Initial Draft
Jan 14	1.1	Minor changes
Jan 16	1.2	team review and revision

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
ML	Machine Learning
UI	User Interface
AI	Artificial Intelligence
AR	Augumented Reality
App	Application
API	Application programming interface
REST	Representational state transfer
RGB	Red, Green, Blue
OS	Operating System
macOS	Operating system developed by Apple Inc
SRS	Software Requirement Specification
VnVPlan	Validation and Verification Plan
MG	Module Guide
MIS	Module Interface Specification

Contents

1 Revision History	i
2 Reference Material	ii
2.1 Abbreviations and Acronyms	ii
3 Introduction	1
4 Purpose	1
5 Scope	1
5.1 System Context Diagram	2
6 Project Overview	2
6.1 Normal Behaviour	2
6.2 Undesired Event Handling	2
6.3 Component Diagram	3
6.4 Connection Between Requirements and Design	4
7 System Variables	5
7.1 Monitored Variables	5
7.2 Controlled Variables	5
7.3 Constants Variables	5
8 User Interfaces	6
9 Design of Communication Protocols	6
10 Timeline	6
A Interface	8
B Communication Protocols	9
C Reflection	9
C.1 Bill Song	9
C.2 Charlotte Cheng	9
C.3 Senni Tan	10
C.4 Marlon Liu	10
C.5 Qiushi Xu	11
C.6 Hongwei Niu	11

List of Tables

1	Design Decisions	4
2	Design Decisions Continued	5
3	Constants Variables	5
4	Timeline	7

List of Figures

1	System Context Diagram	2
2	Component Diagram Diagram	3
3	Hairstyle View of the app, exported from Figma	8
4	App Settings of the app, exported from Figma	8

3 Introduction

The purpose of the project is to deliver a comprehensive, well-documented software application that complies with standards set in SRS, HA, VnVPlan, MG, and MIS.

SRS - [/docs/SRS/SRS.pdf](#)

Hazard Analysis - [/docs/HazardAnalysis/HazardAnalysis.pdf](#)

VnVPlan - [/docs/VnVPlan/VnVPlan.pdf](#)

MG - [/docs/SoftArchitecture/MG.pdf](#)

MIS - [/docs/SoftDetailedDes/MIS.pdf](#)

4 Purpose

The purpose of this document is to effectively communicate the overall design of a system that complies with SRS's specifications. This includes the choices and considerations made by the development team for the system's constituent parts. By the end of this document, the reader ought to understand in detail:

- System-User interactions
- Timing constraints that are present with the system design
- What is considered as normal operation vs. undesirable events that are likely to occur
- Implementation difficulties that may arise as a result of the system design
- Individual system components and their functionality

5 Scope

Many people wonder how a different hairstyle or hair color would look on them in real life. Most of them rely on their imagination and feelings, and some uses some applications with simple filters. Thus, the existing market lacks applications that can accurately simulate their look with a new hairstyle or color.

5.1 System Context Diagram

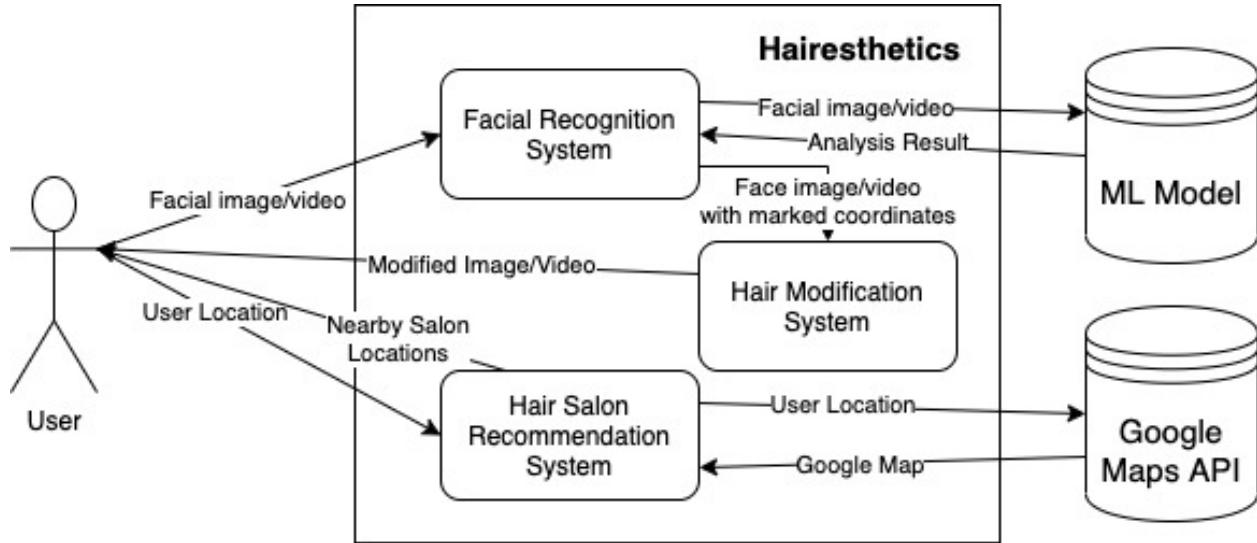


Figure 1: System Context Diagram

6 Project Overview

6.1 Normal Behaviour

Under normal conditions, the system shall perform the desired behaviour and produce the desired response. Desired output include transforming image or video with chosen hair color or hairstyle; output recommended hair salon locations and display them on the map. All requirements mentioned in the SRS document will be met.

6.2 Undesired Event Handling

Encountering any unexpected events and exceptions during runtime, the system shall pause every operations and present Error screen. The Error screen must inform the user that an undesired event or error have occurred and allow the user to redo the last action or re-launch the application. The system will enter the Error screen when one of the following undesired event happened:

1. Internet connection lost during salon recommendation operation.
2. System crash

6.3 Component Diagram

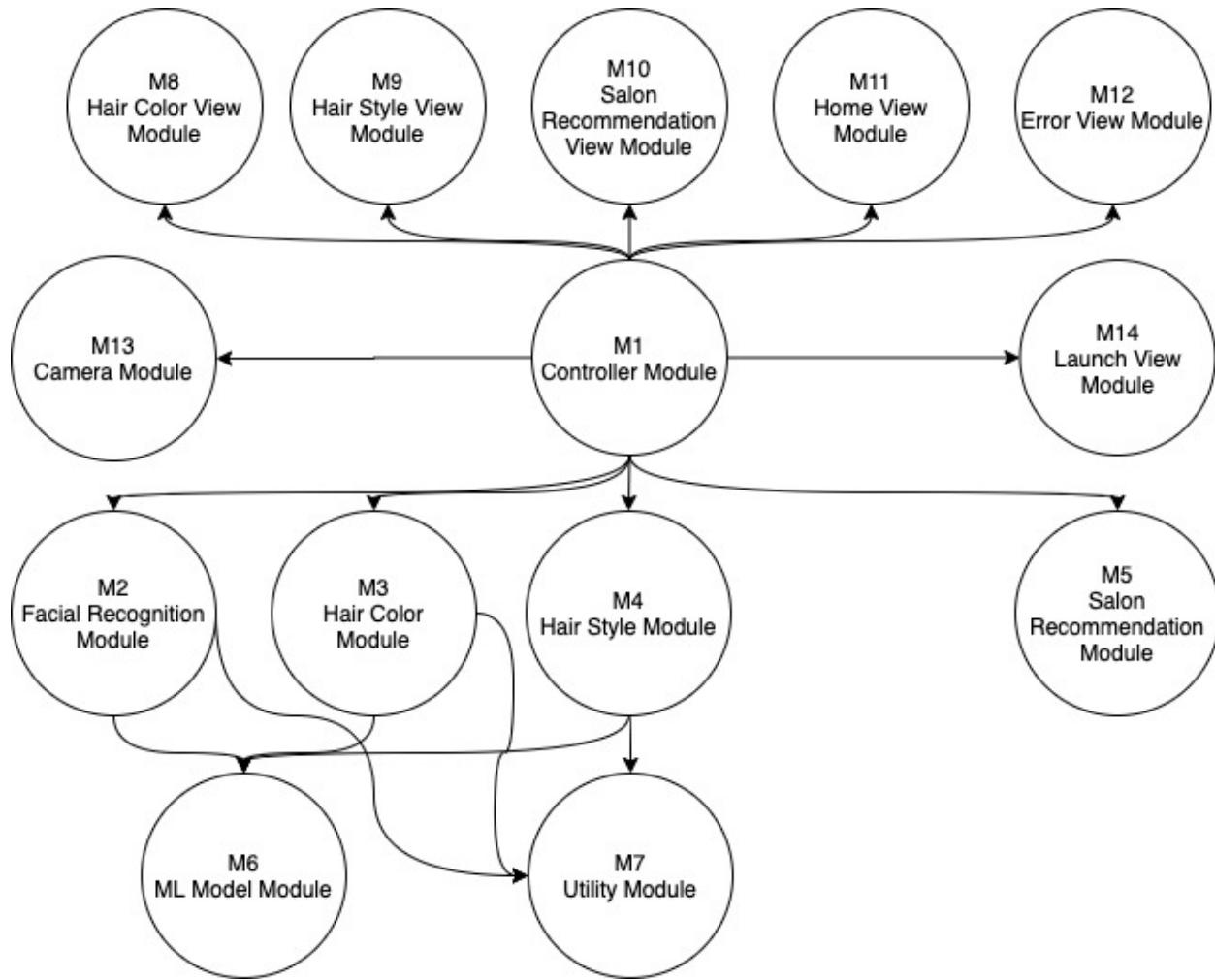


Figure 2: Component Diagram Diagram

6.4 Connection Between Requirements and Design

Table 1: Design Decisions

Requirement Number	Requirement	Design Decision
FR1	The user must input their face in image or video format	Use camera to capture image or video
FR2	The system must pre-process the image by transforming the input facial image to grayscale	Use OpenCV library to convert the image to grayscale
FR3	The system must perform face shape detection on grayscale facial image	Use OpenCV pre-trained facial detection model to detect face shape
FR6	The system must mark the corners and shape with facial landmark coordinates	Use OpenCV facial landmark detection algorithm to mark facial landmarks
FR7	The system must store the coordinates from the hair and facial features	Store the coordinates real-time using cache
HM9	The system should output the transformed face in image or video	Display the change using the camera view
HM10	The system should detect hair edges on detected faces	Use hair segmentation algorithm (UNet) to compute hair edge coordinates
LR1	The installation process must be completed within 3 steps	The app should be available on app store to allow easy installation

Table 2: Design Decisions Continued

Requirement Number	Requirement	Design Decision
ASR1	Fonts shall be in appropriate color and size	Use a predetermined cross-application font family and colors
SER1	The application must allow new features to be added in the future	The application will be designed using a modular architecture
RER1	Data from previous builds of the product should be compatible with new builds of the product	Implement version control for data and backwards compatibility for old data
PRR1	Users will not be able to access data generated by other users	Do not upload user data to cloud OR use authentication system to prevent malicious users from obtaining data

7 System Variables

7.1 Monitored Variables

There are no monitored variables for this system

7.2 Controlled Variables

There are no controlled variables for this system

7.3 Constants Variables

Variable Name	Type	Value	Description
BROWN	Tuple of Integers	[19, 69, 139]	RGB value
PINK	Tuple of Integers	[180, 105, 255]	RGB value
WHITE	Tuple of Integers	[255, 255, 255]	RGB value
PURPLE	Tuple of Integers	[211, 0, 148]	RGB value
GREEN	Tuple of Integers	[113, 179, 60]	RGB value
DARKRED	Tuple of Integers	[0, 0, 139]	RGB value

Table 3: Constants Variables

8 User Interfaces

The user interface for Hairesthetics on Mac OS is designed to be intuitive and easy to navigate. The hairstyle simulation view is divided into three main sections: the camera view, the control panel, and the toolbar. The camera view takes up the majority of the screen and allows the user to view and interact with the simulated hair in real-time. The control panel, located on the lower side of the screen, allows the user to adjust various settings, such as hairstyles and hair colors. The toolbar, located at the top of the screen, provides access to common actions such as app settings.

The app settings for Hairesthetics are accessed through a dedicated settings menu, which is accessible from the toolbar. The settings menu allows the user to configure various aspects of the app, including user preferences and app appearance. It also displays details about application version, privacy levels.

The layout and components of the user interface are detailed in the appendix of this document, which was created using Figma, a design tool.

9 Design of Communication Protocols

Communication protocols are used to define the rules and procedures for data exchange between different modules or systems. In this project, method invocation is used for communication between modules, which allows one module to call a specific function or method in another module. In addition, REST/HTTPS is used as the communication protocol between the module and external APIs. PythonKit framework is used to allow interaction between Swift and Python. As the Python library are loaded at runtime by PythonKit, it will try to find the most modern Python version available in the system. The PythonKit framework allows calling Python script from Swift. This allows for seamless integration and communication between the different modules and external systems, ensuring that data is accurately and efficiently exchanged.

10 Timeline

This is a rough outline of the timeline for the project, and the specific tasks and their duration may change based on the project's needs.

Week	Task	Responsible Member(s)
1-3	Facial feature detection and hair segmentation	Marlon Liu, Bill Song
	UI/UX design and ARKit exploration	Charlotte Cheng, Senni Tan
4-6	API and model training development	Hongwei Niu
	Hair model artist and 3D simulation	Qiushi Xu
7-8	Refining and improving computer vision algorithms	Marlon Liu, Bill Song
	Implementing AR interface in iOS application	Qiushi Xu, Charlotte Cheng, Senni Tan
9-10	Finalizing backend and APIs	Hongwei Niu
	Integrating 3D hair simulations into AR interface	Qiushi Xu
10-13	Final testing, debugging, and polishing	All Members

Table 4: Timeline

A Interface

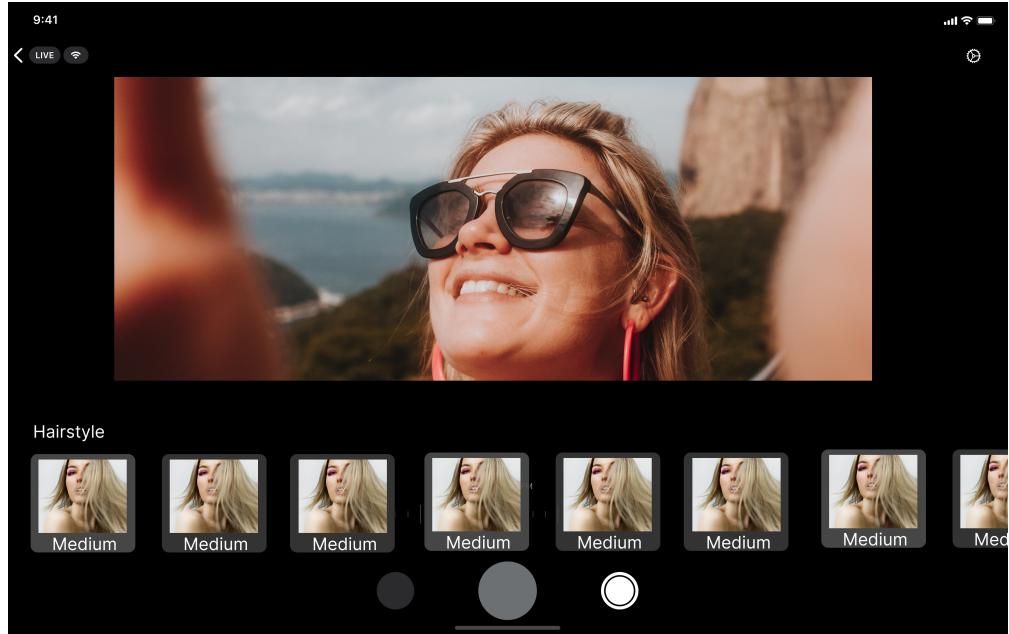


Figure 3: Hairstyle View of the app, exported from Figma

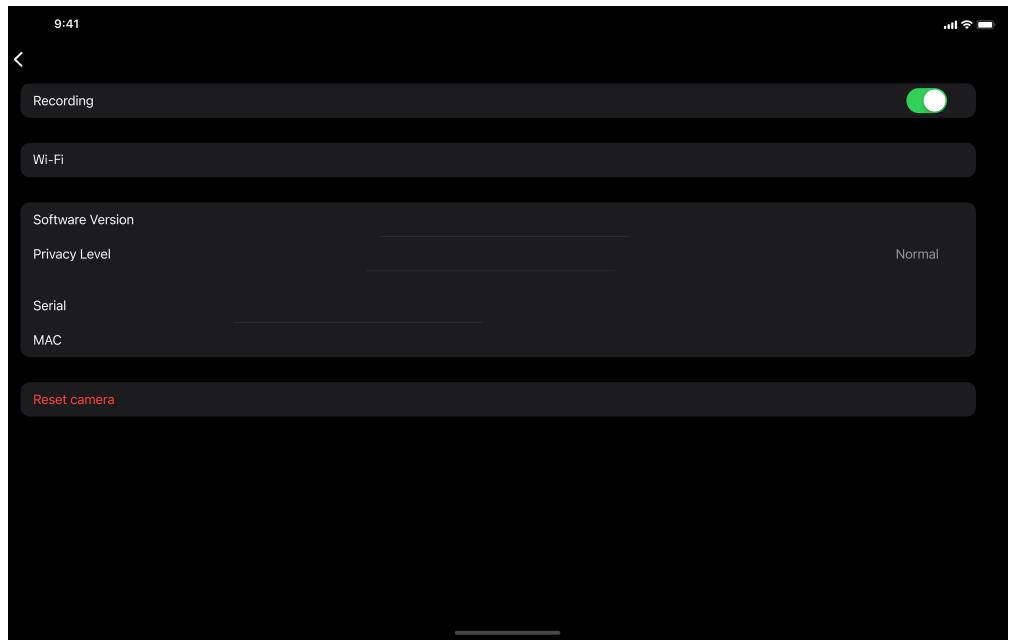


Figure 4: App Settings of the app, exported from Figma

B Communication Protocols

C Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)
2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)

C.1 Bill Song

1. I am responsible for designing and implementing the facial recognition and hair segmentation system. The current design takes into account of the cost and time constraints. Given with unlimited resources, the ML models can achieve a much better accuracy with much more dataset. Finding high quality training dataset consumes a large amount of time, since our system needs picture of different hair color, angle, and style.
2. We chose MVC design pattern for designing the hair modification system. It has separate interfaces and models for changing hair color and hair style. I chose this design, because these modules are mostly independent, so they can be implemented, maintained efficiently in parallel. On the UI side, the interface for these systems share some similarities, MVC allows multiple views and avoid code duplication. During the planning phase, I have considered pip-filter architecture, since hair modification system is essentially a data filter and transformation system. This design supports high throughput for excessively data processing but lacks dynamic interactions and configurations. Hairesthetics should provide interesting and engaging interactions with the user.

C.2 Charlotte Cheng

1. I am responsible for designing the user interface using Figma and implementing it on MacOS using swiftUI. The process of designing the user interface in Figma allowed me to experiment with different layouts and design elements to create a visually appealing and user-friendly interface. However, the transition to implementing the design in SwiftUI was not always smooth. One limitation of my solution for designing the user interface using Figma and implementing it on MacOS using SwiftUI is that it is only compatible with MacOS. Given unlimited resources, I could expand the compatibility

of the application to other operating systems such as Windows and Linux, making it more accessible to a wider audience. Additionally, with more resources, I could conduct more extensive user testing to gather valuable feedback and insights to improve the user experience. Furthermore, I could also explore the use of other design tools and technologies to enhance the visual design and functionality of the application, such as incorporating animations and interactive elements.

2. Another design solution that was considered for this project was implementing the app on iOS. However, this option was not feasible as the PythonKit framework, which is necessary for the integration of Python and Swift, is only compatible with MacOS. One benefit of this design solution is that iOS has a larger user base and more devices available, making the application more accessible to a wider audience. However, the tradeoff is that it would not be able to utilize the PythonKit framework and would require a different method for integration with Python. We eventually decided to go with MacOS app for its compatibility with the PythonKit framework. This allowed for seamless integration with Python and the ability to utilize the full range of functionality provided by the python libraries, which is more important because of the need of Python Machine Learning libraries in the project.

C.3 Senni Tan

1. I am responsible for designing and implementing the camera module. The current design is that the camera is controlled by the Swift package called AVFoundation. This package allows the developer to easily start a record session and currently capture inputs from the physical camera component. My design is that then the session will then pass the camera input to the backend. One limitation is that the performance and reliability of this camera component rely on an outer package so we don't have control of it.
2. Another design solution that we considered is that we can build our own camera component but that is way beyond our experience and knowledge and we may not have time to do this. Also, AVFoundation is a Swift package used by many developers and communities to interact with the backend AR component in their Apps, therefore, this package's performance and reliability are trustable. With all these, we decided to have our design go with using the AVFoundation package.

C.4 Marlon Liu

In summary, I have designed and implemented a facial recognition and hair segmentation system that takes into account cost and time constraints. We have chosen to use the MVC design pattern for our application because it allows for efficient implementation and maintenance of independent modules, it also allows us to follow a test-driven development fashion. I summarized the utility functions into a module to avoid code

duplication and apply a better and useful interface in the higher level development. I also designed the hair segmentation module to integrate the result of machine learning model predictions into image masking. During this time, I have learned the mechanics to design a better backend functionalities and utilized them to serve the frontend better.

C.5 Qiushi Xu

I am responsible for designing and implementing the hairstyle view module and hair color view module. For now the design can put a AR hair model at a certain coordinate in a certain angle. The future design requires the session pass the camera input to the facial recognition process. Then the hairstyle view module and hair color view would receive the face coordinates to place the ar hair model and color. Currently one difficulty would be rotate the AR hair model along with the user's face without huge delay. Since running python code in swift has a lag that may cause some delay, we will simplify the coordinates calculation algorithm to reduce the total run time.

C.6 Hongwei Niu

Using Python as my primary programming language, I set out to create a solution that would allow users to easily find and choose the perfect salon for their needs. Originally, we chose to use a robust database management system to store and manage the data, instead of using an Excel table and pandas library. A database management system such as MySQL or MongoDB would provide more efficient and reliable data management and handling capabilities. However, we choose not to use database system for outputs for this module and other modules. So the outputs currently stored in a spreadsheet. I began by creating a Python script that utilized the pandas library to read an Excel table containing business information and convert it into a DataFrame. Next, I chose use the Google Maps API to extract additional data such as latitude and longitude of the businesses and add them to the DataFrame. With this data in hand, our group used the SwiftUI library to develop a MacOS application that would allow users to easily view and interact with the information. I think implementing a List view to display the business information from the Core Data entities will be reasonable, then use the Map view to display the location of the businesses on the map by plotting the latitude and longitude obtained from the Google Maps SDK. Additionally, We may use the MapKit framework to interact with the Map view, allowing users to zoom, pan, and explore the map view as needed. In the application UI, we may use the data from the Core Data entities and the additional data from the Google Maps SDK to display the information of the businesses and their locations on the map. Users can search, filter and sort the data to find the perfect salon that fits their needs.