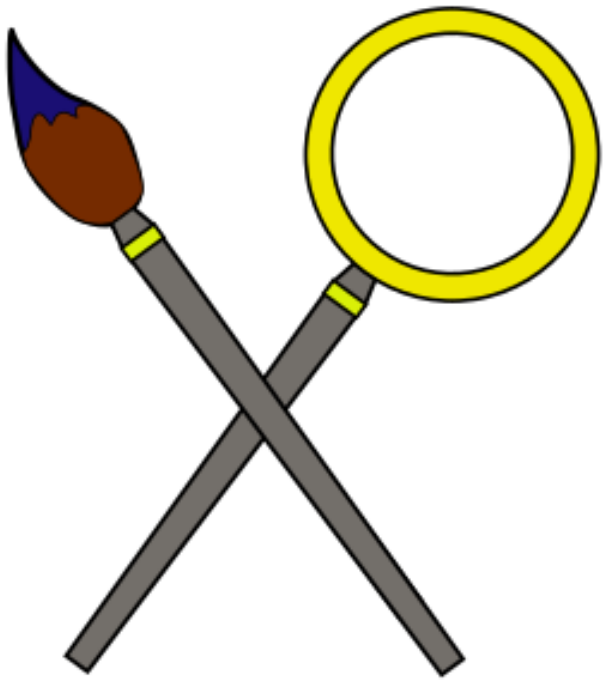
Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Summer 2021



Team Looking Glass

Looking Glass

Michael Aiyedun
Matthew Speer
Suman Raman Poudel
Alexander West

Revision History

Revision	Date	Author(s)	Description
0.1	6.16.2021	MS	document creation
0.2	6.21.2021	AW,SP	porting of content from ADS
0.2	6.22.2021	AW	editing application layer
0.2	6.23.2021	MS	additional sections added
0.2	6.24.2021	AW	revised Cover and Application
0.2	6.27.2021	MS	Final revisions
0.3	8.06.2021	AW	began edits for android changes
0.3	8.07.2021	AW	updated logo and removed obsolete sections
0.3	8.11.2021	AW	began re-writing sections

CONTENTS

List of Figures

1 Introduction

Looking Glass functions as a QR code reading software for smart glasses, to provide users with the ability to access stories and information on QR codes. Looking Glass is installed on smart glasses by museums to scan QR codes placed around their exhibits in order to offer additional context and visual and audio mediums to learn about the art installations. It allows exhibit owners to create their own QR codes and customize the information they provide to their patrons.

2 System Overview

Looking Glass will function under five primary layers with one of them, the application layer, being the central unit that connects all the others. The application layer will handle all the input from the interface, camera, motion input, and data layers as well as make sure to return the necessary output for each different layer to properly function. The interface layer will be one of two primary user layers, managing the user experience. The camera layer will be the other user interactive layer, following the user's directive to determine what information they wish to receive. The motion input layer will be a secondary back end system that will determine the user's intent based on their head movement. The data layer will store the content necessary for exhibits to archive and be retrieved for the user upon requests received from the application layer.

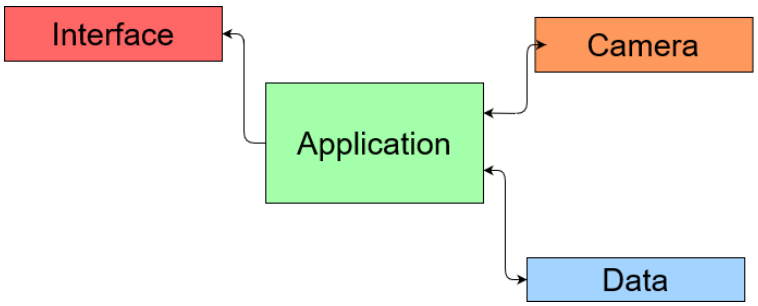


Figure 1: Looking Glass layer overview

2.1 Interface Layer Description

This layer is responsible for the main bulk of the user's interaction with the application. It will feature the audio, video, and text for exhibits that the user wishes to view as well as be responsible for input such as the touch controls.

2.2 Camera Layer Description

The camera layer will receive permission from the application layer to begin scanning for QR codes. It takes an image for the scan and the program parses the image for a QR code.

2.3 Application Layer Description

The application layer holds the bulk of the software as it takes all the requests from all other layers and outputs them accordingly to ensure proper processing. It will receive the touch inputs from the interface layer to determine application state and send the proper audio and visuals the user wishes to experience. To the camera layer, it will take an image from the camera user's behest and then it scans for a qr code and will search the linked list for that entry.

2.4 Data Layer Description

The data layer will store all the audio, video, and text for each exhibit. Upon request from the application layer, it will retrieve the necessary data and send it back to the application layer to be processed.

3 Subsystem Definitions & Data Flow

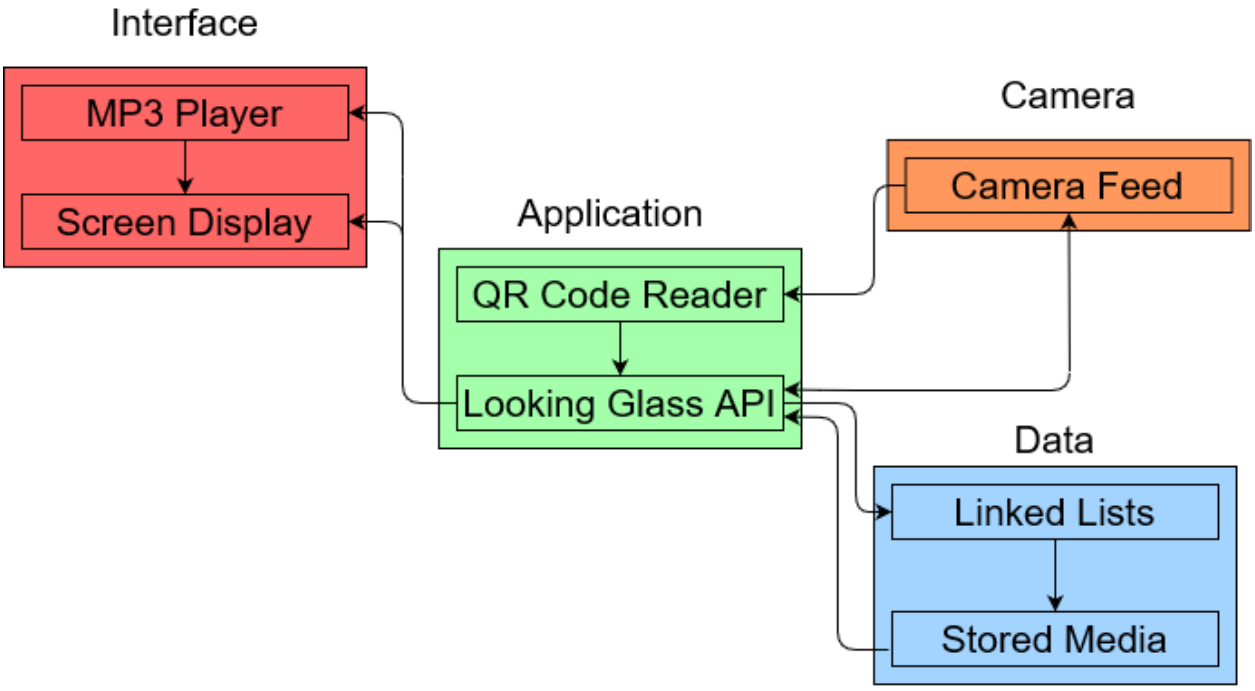


Figure 2: Looking Glass data flow diagram

4 Application Layer Subsystems

The application layer is the core of Looking Glass. It is responsible for the interaction of all other layers as it receives their output and provides the input necessary for them to function. It is composed of just the Looking Glass API.

4.1 Application Layer Hardware

The Looking Glass Application is intended to be used on a google glass platform.

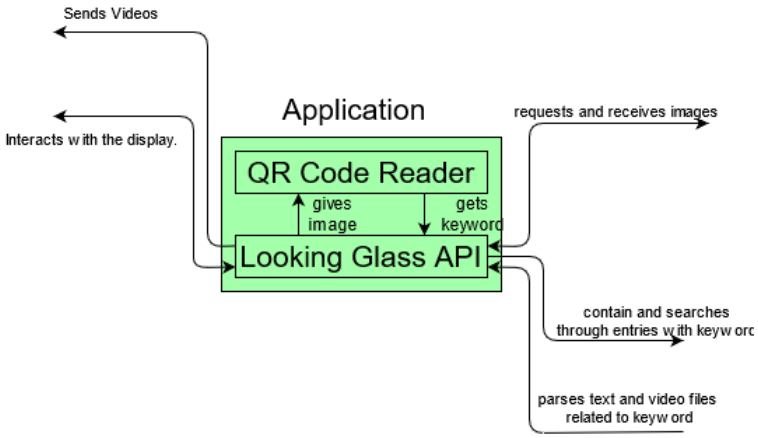


Figure 3: Looking Glass API diagram

4.2 Application Layer Operating System

The Looking Glass Application is designed to run on the Google Glass OS. The Glass system is an Android based OS.

4.3 Application Layer Software Dependencies

Dependencies have not been determined.

4.4 Looking Glass API

The Looking Glass API will handle the entirety of the application flow. It will receive the input from the interface layer and return the interface contents necessary to the viewer. The API will allow the camera to activate and analyze its contents. In addition it will send out data requests to the data layer and receive the stored media to return to the interface layer.

4.4.1 Subsystem Hardware

It will run on an Android OS Smartphone.

4.4.2 Subsystem Operating System

It will run on an Android OS.

4.4.3 Subsystem Software Dependencies

It depends on a Java version of at least Java8.

4.4.4 Subsystem Programming Languages

It is coded in Java.

4.4.5 Subsystem Data Structures

The API stores the exhibits text and video within the program.

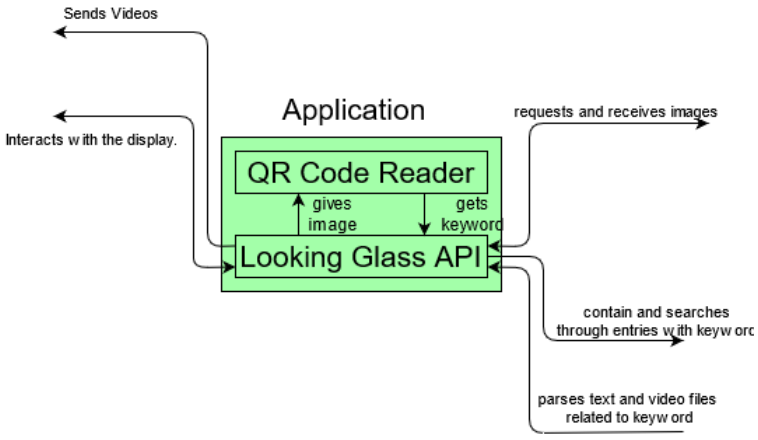


Figure 4: Looking Glass API diagram

4.5 QR Code Reader

The QR Code Reader is a function called in the main activity of the program every time a scan is run.

4.5.1 Subsystem Hardware

It will run on an Android OS Smartphone.

4.5.2 Subsystem Operating System

It will run on an Android OS.

4.5.3 Subsystem Software Dependencies

It depends on a Java version of at least Java8.

5 Data Layer Subsystems

The data layer holds the information relevant for exhibits stored away for the user to access on a whim. It receives the QR code request from the application layer upon which it will access the proper stored media and then return it to the application to present to the user. It is composed of a QR code collection and stored media.

5.1 Data Layer Hardware

The hardware required for Looking Glass will be a pair of smart glasses as it will contain all necessary data stored directly in the smart glasses.

5.2 Data Layer Operating System

The data will operate on the Google Glass OS which is based on a version of Android OS.

5.3 Data Layer Software Dependencies

Currently there are no planned software dependencies for the data layer.

5.4 Linked List

The linked list is the storage of the text for the exhibits, each one tied with a keyword and having links to videos. Each entry in the linked list is tied to a plaintext keyword that the QR code will give such as "monalisa" or "shakespeare".

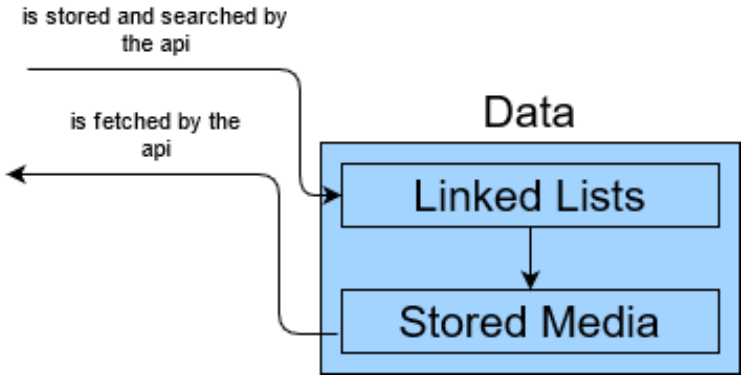


Figure 5: QR Codes diagram

5.4.1 Subsystem Hardware

Will require Google Glass.

5.4.2 Subsystem Operating System

Will operate on the Google Glass OS.

5.4.3 Subsystem Software Dependencies

Will need to include dependencies based off QR code structure.

5.4.4 Subsystem Programming Languages

The programming language has not been determined.

5.4.5 Subsystem Data Structures

Each QR code will be linked to a data structure of stored media.

5.4.6 Subsystem Data Processing

There is no algorithmic approach planned for the QR codes.

5.5 Stored Media

The owner will be able to upload and remove their desired content here and then have QR codes assigned to the data to send to the interface layer ultimately to be viewed by the user.

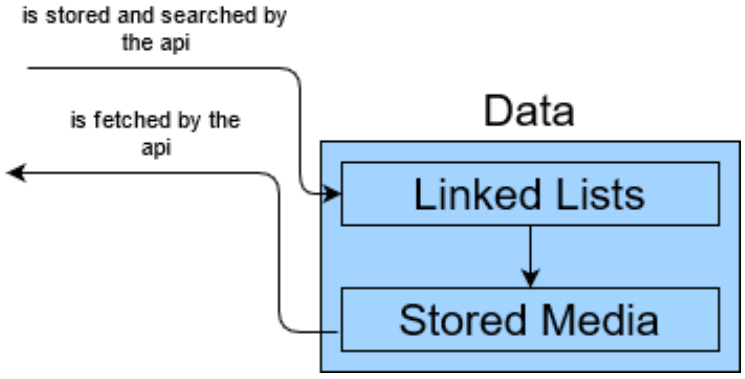


Figure 6: QR Codes diagram

5.5.1 Subsystem Hardware

It will run on an Android OS Smartphone.

5.5.2 Subsystem Operating System

It will run on an Android OS.

5.5.3 Subsystem Software Dependencies

It depends on a Java version of at least Java8.

5.5.4 Subsystem Programming Languages

It is coded in Java.

5.5.5 Subsystem Data Structures

Stored media data structures will contain a text, audio, and video node attachment.

5.5.6 Subsystem Data Processing

There is no algorithmic approach planned for the stored media.

6 Interface Layer Subsystems

The interface layer primarily deals with the encapsulation of the application, presenting the audio and visual information to the user while keeping track of the physical input as well. The interface layer is divided into the touch input, mp3 player, and screen display.

6.1 MP3 PLAYER

The MP3 player will be responsible for playing all mp3 files for any audio content. It will receive the necessary information from the pipeline of the application layer retrieving audio from the data layer.

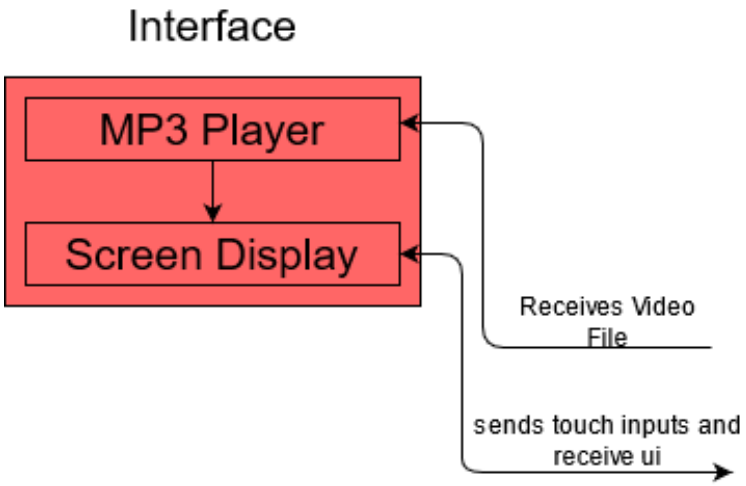


Figure 7: MP3 Player diagram

6.1.1 Subsystem Hardware

It will run on an Android OS Smartphone.

6.1.2 Subsystem Operating System

It will run on an Android OS.

6.1.3 Subsystem Software Dependencies

It depends on a Java version of at least Java8.

6.1.4 Subsystem Programming Languages

It is coded in Java.

6.1.5 Subsystem Data Processing

a long running context to own them for the entire time that they are visible, so managing them in a background service. Using Low and High Frequency Rendering

6.2 Screen Display

The screen display will present the user with all text and video content associated with any scanned QR code. It will act as the primary method of interaction with the user and providing them with feedback of the application.

6.2.1 Subsystem Hardware

The hardware will vary from smartphone to smartphone.

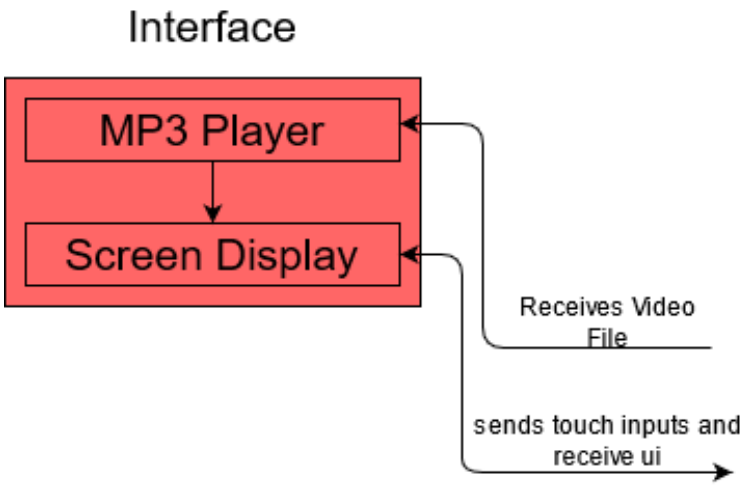


Figure 8: MP3 Player diagram

7 Camera Layer Subsystems

The camera layer is responsible for providing visual data and scanning for QR codes for the application to process. It is composed of a QR code reader and a camera feed.

7.1 Camera Layer Hardware

The camera varies from phone to phone.

7.2 Camera Layer Operating System

The cameras should all be interacted with through the android OS.

7.3 Camera Layer Software Dependencies

There are no planned software dependencies.

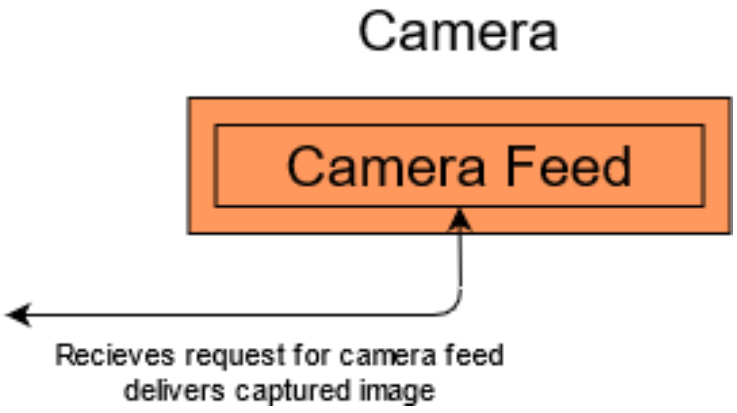


Figure 9: QR Code Reader diagram

7.4 Camera Feed

The camera feed will handle the requests from the application to begin scanning for QR codes and provide data back to the application layer to determine its desire prediction analysis.

7.4.1 Subsystem Hardware

The camera varies from phone to phone.

7.4.2 Subsystem Operating System

The cameras should all be interacted with through the android OS.