

# Software Requirements Specification for Software Engineering: subtitle describing software

Team 5, GradSight

Willie Pai

Hammad Pathan

Wajdan Faheen

Henushan Balachandran

Zahin Hossain

March 29, 2025

# Contents

<b>1</b>	<b>Purpose of the Project</b>	<b>vi</b>
1.1	User Business . . . . .	vi
1.2	Goals of the Project . . . . .	vi
<b>2</b>	<b>Stakeholders</b>	<b>vi</b>
2.1	Client . . . . .	vi
2.2	Customer . . . . .	vi
2.3	Other Stakeholders . . . . .	vii
2.4	Hands-On Users of the Project . . . . .	vii
2.5	Personas . . . . .	vii
2.6	Priorities Assigned to Users . . . . .	viii
2.7	User Participation . . . . .	viii
2.8	Maintenance Users and Service Technicians . . . . .	viii
<b>3</b>	<b>Mandated Constraints</b>	<b>viii</b>
3.1	Solution Constraints . . . . .	viii
3.2	Implementation Environment of the Current System . . . . .	ix
3.3	Partner or Collaborative Applications . . . . .	ix
3.4	Off-the-Shelf Software . . . . .	ix
3.5	Anticipated Workplace Environment . . . . .	x
3.6	Schedule Constraints . . . . .	x
3.7	Budget Constraints . . . . .	x
3.8	Enterprise Constraints . . . . .	x
<b>4</b>	<b>Naming Conventions and Terminology</b>	<b>xi</b>
<b>5</b>	<b>Relevant Facts and Assumptions</b>	<b>xiii</b>
5.1	Relevant Facts . . . . .	xiii
5.2	Business Rules . . . . .	xiii
5.3	Assumptions (Continued) . . . . .	xiv
<b>6</b>	<b>The Scope of the Work</b>	<b>xiv</b>
6.1	The Current Situation . . . . .	xiv
6.2	The Context of the Work . . . . .	xiv
6.3	Specifying a Business Use Case (BUC) . . . . .	xvi
6.4	Likely and Unlikely Changes . . . . .	xviii

<b>7</b>	<b>Business Data Model and Data Dictionary</b>	<b>xix</b>
7.1	Business Data Model . . . . .	xix
7.2	Data Dictionary . . . . .	xxi
<b>8</b>	<b>The Scope of the Product</b>	<b>xxii</b>
8.1	Product Boundary . . . . .	xxii
8.2	Product Use Case Table . . . . .	xxiii
8.3	Individual Product Use Cases (PUC's) . . . . .	xxiv
<b>9</b>	<b>Functional Requirements</b>	<b>xxiv</b>
<b>10</b>	<b>Look and Feel Requirements</b>	<b>xxv</b>
10.1	Appearance Requirements . . . . .	xxv
10.2	Style Requirements . . . . .	xxv
<b>11</b>	<b>Usability and Humanity Requirements</b>	<b>xxvi</b>
11.1	Ease of Use Requirements . . . . .	xxvi
11.2	Personalization and Internationalization Requirements . . . . .	xxvi
11.3	Learning Requirements . . . . .	xxvi
11.4	Understandability and Politeness Requirements . . . . .	xxvi
11.5	Accessibility Requirements . . . . .	xxvi
<b>12</b>	<b>Performance Requirements</b>	<b>xxvii</b>
12.1	Speed and Latency Requirements . . . . .	xxvii
12.2	Safety-Critical Requirements . . . . .	xxvii
12.3	Precision or Accuracy Requirements . . . . .	xxvii
12.4	. . . . .	xxvii
12.5	Capacity Requirements . . . . .	xxvii
12.6	Scalability or Extensibility Requirements . . . . .	xxviii
12.7	Longevity Requirements . . . . .	xxviii
<b>13</b>	<b>Operational and Environmental Requirements</b>	<b>xxviii</b>
13.1	Expected Physical Environment . . . . .	xxviii
13.2	Wider Environment Requirements . . . . .	xxviii
13.3	Requirements for Interfacing with Adjacent Systems . . . . .	xxix
13.4	Productization Requirements . . . . .	xxix
13.5	Release Requirements . . . . .	xxx

<b>14 Maintainability and Support Requirements</b>	<b>xxxix</b>
14.1 Maintenance Requirements . . . . .	xxxix
14.2 Supportability Requirements . . . . .	xxxix
14.3 Adaptability Requirements . . . . .	xxxix
<b>15 Security Requirements</b>	<b>xxxix</b>
15.1 Access Requirements . . . . .	xxxix
15.2 Integrity Requirements . . . . .	xxxix
15.3 Privacy Requirements . . . . .	xxxix
15.4 Audit Requirements . . . . .	xxxix
15.5 Immunity Requirements . . . . .	xxxix
<b>16 Cultural Requirements</b>	<b>xxxix</b>
<b>17 Compliance Requirements</b>	<b>xxxix</b>
17.1 Legal Requirements . . . . .	xxxix
17.2 Standards Compliance Requirements . . . . .	xxxix
<b>18 Open Issues</b>	<b>xxxix</b>
<b>19 Off-the-Shelf Solutions</b>	<b>xxxix</b>
19.1 Ready-Made Products . . . . .	xxxix
19.2 Reusable Components . . . . .	xxxix
19.3 Products That Can Be Copied . . . . .	xxxix
<b>20 New Problems</b>	<b>xxxix</b>
20.1 Effects on the Current Environment . . . . .	xxxix
20.2 Effects on the Installed Systems . . . . .	xxxix
20.3 Potential User Problems . . . . .	xxxix
20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	xxxix
20.5 Follow-Up Problems . . . . .	xxxix
<b>21 Tasks</b>	<b>xl</b>
21.1 Project Planning . . . . .	xl
<b>22 Planning of the Development Phases</b>	<b>xl</b>
<b>23 Migration to the New Product</b>	<b>xl</b>

24 User Documentation and Training	xliii
25 Waiting Room	xliv
26 Ideas for Solution	xliv
27 Traceability Matrix	xlvi

## Revision History

Date	Version	Notes
Date 1	1.0	Notes
03/29/2025	1.1	Revision for Final Submission

# **1 Purpose of the Project**

## **1.1 User Business**

The GradSight project is designed to modernize the way graduation composites are displayed and managed within McMaster University’s Faculty of Engineering. Currently, physical composites are difficult to access and maintain, hindering the tool to alumni, students, and faculty. By creating a digital, scalable platform, our project aims to provide an accessible, user-friendly system for browsing, searching, and interacting with graduate composites. This system is intended to improve the user experience, allowing for access to historical and future composites, while ensuring security and privacy.

## **1.2 Goals of the Project**

The primary goals of the project are to develop a functional and scalable digital application that provides composite management for the Faculty of Engineering. The project focuses on creating a long-term solution that can handle the needs of both current and future graduating classes. Additionally, our team aims to implement an intuitive user experience with robust data management, including the integration of Optical Character Recognition (OCR), if necessary, for digitizing physical composites.

# **2 Stakeholders**

## **2.1 Client**

### **McMaster University Faculty of Engineering**

The client for the project is the McMaster University Faculty of Engineering. This will include the in-house development of a digital composite display system to help engage students and alumni through a searchable, interactive database of alumni graduation composites.

## **2.2 Customer**

### **Alumni Relations Department**

This would be the Alumni Relations Department, which is responsible for ensuring that the system provides everything the alumni needs to keep their goals related

to engagement and data privacy and integrity. They will also provide continuous feedback to the development team to keep them in sync with their objectives.

## 2.3 Other Stakeholders

### **Dr. Spencer Smith (Capstone Supervisor)**

Oversees the project academically to make sure it complies with capstone criteria and maintains a high level of technical and instructional value.

### **Cindy Ashford (Lifetouch)**

Lifetouch is a vendor that will be creating and providing the composites to the university. They may have some involvement in the delivery of the alumni photos for the digital interface. Cindy Ashford represents Lifetouch in regards to reviewing security and privacy concerns about the use of these photos.

## 2.4 Hands-On Users of the Project

### **Students and Alumni**

These are the major end-users who will interact with the touchscreens in their search and viewing of alumni composite albums. They would need an intuitive, easy-to-navigate interface to search by name, program, or year.

### **General McMaster Community**

All of the general McMaster community (faculty, staff, visitors, and prospective students) are stakeholders and potential hands-on users of the project. Given that the touchscreen kiosks will be publicly accessible on campus, any one person at McMaster could be interacting with the system. Due to the nature, in that they can use the displays to explore the digital composites, one should also consider their user experience, though they will not be the primary target audience, such as alumni or current students.

## 2.5 Personas

**Engineering Students:** Utilize the system to determine where graduates are, what they are doing, explore career options, and keep current with alumni.

**Alumni:** Use the system to access their composites, maintain peer networking, and update professional information in case additional functionality is added.

**Faculty/Staff:** The system should be used to feature alumni success in recruitment events or when hosting prospective students.



## 2.6 Priorities Assigned to Users

**Primary Users - High Priority:** Include alumni and current students who will use the system most frequently.

**Secondary Users - Medium Priority:** Faculty and staff as they communicate through the system about alumni engagement.

**Tertiary Users - Low Priority:** Maintenance staff, who keep the system running and updated.

## 2.7 User Participation

**Alumni Relations Department:** Ongoing feedback on behalf of development and deployment with regards to how the system is meeting the needs of alumni and standards for privacy.

**Usability Testing by End Users:** A small number of students and alumni will be involved in usability testing regarding interface and navigation and overall user experience.

## 2.8 Maintenance Users and Service Technicians

**The IT and Maintenance Staff:** These are personnel who, after deployment, do all the technical work to update software and deal with the maintenance of the system. They need administrative tools to understand and troubleshoot problems as fast as possible.

**System Admins:** Their job is to keep a back-end database management system updated, ensuring security but also updating and modifying the information in the system whenever needed.

# 3 Mandated Constraints

## 3.1 Solution Constraints

- **Data Security & Privacy:** The system will adhere to McMaster University's policy on data protection, since information dealing with alumni is sensitive. Alumni information stored and processed in the system should be encrypted, and access must be granted to authorized users only.

- **Touchscreen Compatibility:** The system should be optimized for different types of hardware, ranging from larger displays down to iPads and tablets, to provide a seamless user experience across hardware.
- **Limited in External Access:** The system will be running over McMaster University's internal servers, which give absolutely no access from the outside to on-campus use only.

### 3.2 Implementation Environment of the Current System

**Hardware:** The system resides on McMaster University's large touchscreen displays. It should also be accessible on smaller devices, such as iPads and other types of tablets.

**Software:** The project will implement a responsive web application for a great experience on different screen sizes. It will be hosted on McMaster Internal Servers, and data will be securely managed on an encrypted database.

**Operating Systems:** It should be running a Linux-based OS on the backend on mini-PCs; for a web application, it must work on multiple platforms like iOS and Android.

### 3.3 Partner or Collaborative Applications

**Lifetouch:** It can provide composite images of alumni. The system should integrate with the image repositories at Lifetouch via a secure mechanism. It shall respect all privacy agreements.

**University Systems:** The project may use existing McMaster IT infrastructure for data storage, user authentication, and network access.

### 3.4 Off-the-Shelf Software

**Version Control:** It provides version control to maintain the issue-tracking history and thus helps the team work together smoothly.

**Web Frameworks:** The responsive web application will be built using off-the-shelf web frameworks.

**DBMS:** Alumni data, photos, and logs about the activity of its registered users will be stored in the database.

### 3.5 Anticipated Workplace Environment

**Development Environment:** Most of the work would be in a hybrid model; some work will be completely remote through Microsoft Teams, while hands-on things, such as setting up hardware and testing, will be in-person.

**Campus Deployment:** The system will be installed in appropriate locations at McMaster University, 2nd and 3rd floors of JHE, and near the Alumni and Recruitment office. Regular communication with on-campus IT services and facilities management will be needed.

### 3.6 Schedule Constraints

**Academic Timeline:** The project is to be concluded strictly in compliance with the university's capstone schedule, including milestones like mid-term evaluations, prototype demonstrations, and final presentations.

**Hardware Availability:** Delays in acquiring necessary hardware-touchscreen displays and mini-PCs may affect the timeline of implementation.

### 3.7 Budget Constraints

**Hardware Costs:** The system will be very expensive in terms of touchscreen displays and mini-PCs or any other similar hardware. The costs may have to be restricted by selecting budget-friendly alternatives such as Raspberry Pi.

**Licensing of Software:** If proprietary software is needed, licensing can cut into the bottom line. Open-source alternatives will be favored wherever possible to keep costs lower.

### 3.8 Enterprise Constraints

**University IT Policies:** The system should not violate the IT policies at McMaster University, such as data security, server usage, and software installation protocols.

**Scalability:** It should be designed such that it would be easy to scale up in case of additional touchscreen displays throughout campus.

**Maintenance:** This will be continuing, done by university IT staff. It should be well-documented and relatively easy to maintain without ongoing developer involvement.

## 4 Naming Conventions and Terminology

### Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

- **GradSight:** The name of the capstone project.
- **OCR (Optical Character Recognition):** A technology used in the project to convert physical graduation composites into digital, searchable formats by extracting text (e.g., student names) from scanned images.
- **CI/CD (Continuous Integration/Continuous Deployment):** A development practice where code changes are automatically tested and deployed.
- **Pull Request (PR):** A Git feature used to propose changes to the codebase.
- **Proof of Concept (POC):** An early prototype of the application.
- **API:** Application Programming Interface – a set of rules allowing different systems to communicate with each other.
- **GDPR:** General Data Protection Regulation – European data privacy regulation that your system must comply with.
- **RBAC:** Role-Based Access Control – a security method that restricts system access based on user roles.
- **FR:** Functional Requirement – a specific system behavior or function that must be implemented.
- **NFR:** Nonfunctional Requirement – system qualities like performance, security, usability, etc.
- **PR:** Productization Requirement – constraints or requirements related to deployability, modularity, or setup.
- **RR:** Release Requirement – conditions that must be met before releasing the system.
- **SUP:** Supportability Requirement – requirements regarding maintenance, documentation, or support.

- **ADP:** Adaptability Requirement – system’s ability to adapt to changes in hardware or use case.
- **SEC:** Security Requirement – protections against threats, including access control and encryption.
- **INT:** Integrity Requirement – mechanisms for maintaining data accuracy and trustworthiness.
- **PRIV:** Privacy Requirement – how the system will protect personal data and ensure user consent.
- **AUD:** Audit Requirement – requirements related to system monitoring, logging, and compliance tracking.
- **CUL:** Cultural Requirement – needs related to diverse user backgrounds and language handling.
- **LGL:** Legal/Compliance Requirement – adherence to laws, licenses, and industry standards.
- **FMEA:** Failure Mode and Effects Analysis – a structured table identifying system failure risks and mitigations.
- **SRS:** Software Requirements Specification – this document.
- **DBMS:** Database Management System – the software layer used for storing and retrieving data.
- **UI/UX:** User Interface / User Experience – design considerations for how users interact with the system.
- **DoS:** Denial of Service – a cyber attack that disrupts system access by overwhelming it with traffic.
- **XSS:** Cross-Site Scripting – a type of security vulnerability allowing code injection via user inputs.
- **HTTPS/TLS:** HyperText Transfer Protocol Secure / Transport Layer Security – secure protocols for encrypted web communication.

## 5 Relevant Facts and Assumptions

### 5.1 Relevant Facts

The target users include alumni, current students, and faculty members who require an intuitive interface to browse and search through digital class composites.

The system is expected to run on interactive touchscreen displays located within McMaster University buildings, as well as be accessible via desktop interfaces.

The project stakeholders include:

- The Faculty of Engineering (project sponsor)
- Lifetouch (external photography provider)
- McMaster University IT Services (system host and post-deployment support)

The system will be developed using an agile approach, with development roles designated within the team to streamline collaboration. Responsibilities include note-taking, client communication, quality assurance, and documentation review.

A modular architecture will support maintainability and scalability across multiple user-facing devices and data sources.

### Assumptions

- McMaster University's IT infrastructure will support secure deployment, with access to internal servers and campus network resources.
- Lifetouch will provide access to digital composites or APIs for integration.
- Final technology decisions (such as frameworks, databases, or testing tools) will be made based on evolving requirements and institutional compatibility standards.
- All composite data and related media provided by external vendors will adhere to privacy and intellectual property policies applicable to McMaster University.

### 5.2 Business Rules

**User Access:** Only authorized users (alumni, students, faculty) will be able to browse, search, and interact with the digital composites.

**Intellectual Property (IP) Compliance:** The project must respect intellectual property rights, specifically regarding the images provided by LifeTouch.

**Search and Browsing Features:** The system must allow users to search composites by year, faculty, or individual names, with filters available to narrow down results.

**Data Privacy and Security:** The system must comply with university privacy policies, ensuring that personal data (such as student names and photos) are securely stored and protected from unauthorized access.

### 5.3 Assumptions (Continued)

**Availability of Digital Images:** It is assumed that LifeTouch will provide the required digital versions of past composites in a timely manner and with sufficient image quality for OCR to function effectively.

**OCR Reliability:** The Optical Character Recognition (OCR) technology is assumed to work effectively in converting printed composites into digital formats, with minimal manual intervention required. In case of OCR errors, manual entry will be a fallback solution.

**Internet Accessibility:** Users will have consistent access to the internet to browse and search the digital composites, as the platform is expected to be hosted online and not available offline.

## 6 The Scope of the Work

### 6.1 The Current Situation

The current situation is that there are physical composites around the campus, specifically in the JHE building. These photographs are framed and placed on the wall of the buildings, scattered. Additionally, with more composites arising as years go by, there becomes limited space in the JHE building to accommodate engineering composites.

### 6.2 The Context of the Work

The system is intended to run on touchscreen displays located on the second and third floors of the John Hodgins Engineering (JHE) building at McMaster University. It allows users (students, alumni, and faculty) to browse, search, and view composite images of graduating students. The backend is hosted on McMaster's internal servers,

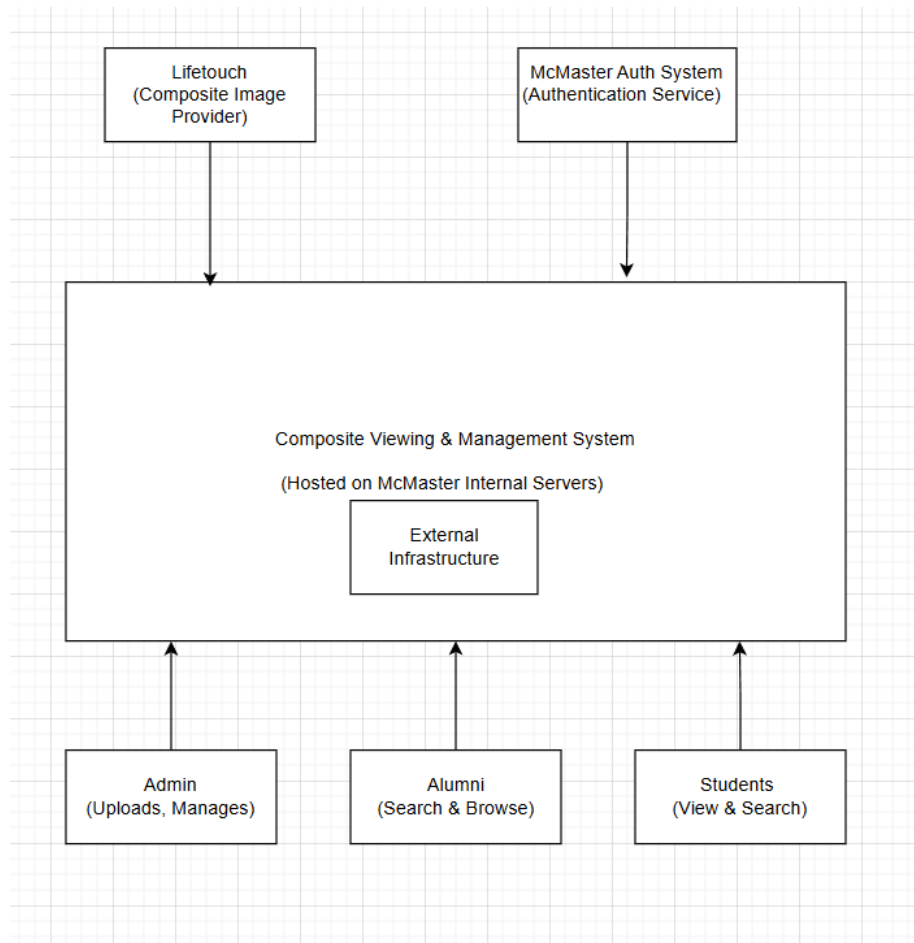


Figure 1: System Context Diagram for Composite Viewing and Management

with integrations to external systems for photo data and identity verification. The context diagram below illustrates the primary external systems, user roles, and their interactions with the system.



Work can be partitioned into well-defined tasks, focusing on both technical and management aspects. The project can be broken down into the major frameworks of the product. This can be specified as Front-end development (visual application and interactions), back-end development (control interactions and provide the content), Database management (composite datastore), OCR integration (breakdown composites and extract information), and finally testing and quality assurance to ensure the product is bug-free and scalable for real-world deployment.

## 6.3 Specifying a Business Use Case (BUC)

**Use Case Name:** View and Search Graduation Composite

**Primary Actor:** Alumni, Students, Faculty Members

### **Stakeholders and Interests:**

- **Alumni:** Want to easily access and view their graduating class composite and search for classmates by name.
- **Students:** Want to browse past composites, potentially finding connections or learning more about the institution's history.
- **Faculty:** Need a streamlined way to manage and update composite information across various graduating classes and make it accessible to students and alumni.

### **Preconditions:**

- The user has access to the GradSight platform.
- The user has a working internet connection.
- The graduation composite data (names, photos) is already digitized and available in the system.

### **Postconditions:**

- The user successfully views the composite of a chosen graduating class.
- The user is able to search for individuals by name or filter by graduation year, program, or faculty.

- The user can interact with the displayed composite, zooming in for detailed views and accessing additional information (e.g., individual names and related details).

### **Main Success Scenario (Basic Flow):**

1. **Login:** The user navigates to the GradSight platform and logs in with their credentials (if applicable).
2. **Composite Selection:** The user selects the “Browse Composites” option from the menu. The system displays a list of available graduating classes, categorized by year, faculty, or department.
3. **View Composite:** The user selects a specific graduating class (e.g., 2020 Engineering). The system retrieves and displays the digital composite image for the selected year, along with a searchable interface.
4. **Search by Name:** The user enters a name or keyword into the search bar. The system performs an Optical Character Recognition (OCR)-based search on the composite data and highlights the individual matching the search criteria. The user is presented with the corresponding image, name, and additional relevant details (e.g., program or title).
5. **Interactive View:** The user zooms in on the composite to explore more detailed sections. The user can click on individual faces to access additional data (e.g., name, program, year).
6. **Exit:** The user finishes their session and logs out of the system.

### **Extensions (Alternate Flows):**

- **Composite Not Found:** If the selected graduating class composite is not available in the system, the user receives a message indicating that it is under processing or not yet digitized. The system may offer an option to notify the user when it becomes available.
- **Search Not Found:** If no matching name is found for the search query, the system notifies the user that the individual is not in the selected composite or suggests other relevant results.

## 6.4 Likely and Unlikely Changes

Understanding which aspects of the system are likely to change versus those that are expected to remain stable is essential for future development, testing, and maintenance. The following categorization helps prioritize flexibility in design and long-term planning:

### Likely Changes

- **Hardware Models:** The specific touchscreen displays and mini-PCs may be replaced with newer or more cost-effective alternatives (e.g., updated Raspberry Pi models). The system should remain compatible with a range of hardware.
- **User Interface Design:** The layout, theme, and structure of the user interface may be modified based on user feedback, accessibility needs, or branding updates from the university.
- **Search and Browsing Features:** Filters, sorting mechanisms, and result presentation in the alumni and student view pages may evolve to improve usability or support additional metadata.
- **Deployment Locations:** While the system is initially targeted for specific areas (e.g., 2nd and 3rd floors of JHE), future installations across campus may be introduced as the system scales.

### Unlikely Changes

- **Privacy Requirements:** Compliance with McMaster University’s data handling and privacy protocols, especially concerning alumni images, is expected to remain consistent and foundational.
- **Authentication System:** Integration with McMaster’s internal authentication infrastructure is a stable, mandated constraint and not expected to change.
- **Access Control Logic:** Role-based access management for students, alumni, and administrators is central to the system’s architecture and will likely remain unchanged.
- **On-Campus Hosting:** Given university policy on internal data access and server usage, external hosting is not anticipated unless institutional policies change.

## 7 Business Data Model and Data Dictionary

### 7.1 Business Data Model

- **User to Student:**  
One-to-One (1:1)  
Each user can be a student, and each student corresponds to one user account.
- **GraduationComposite to Student:**  
One-to-Many (1:N)  
Each graduation composite can contain multiple students, but each student belongs to only one composite.

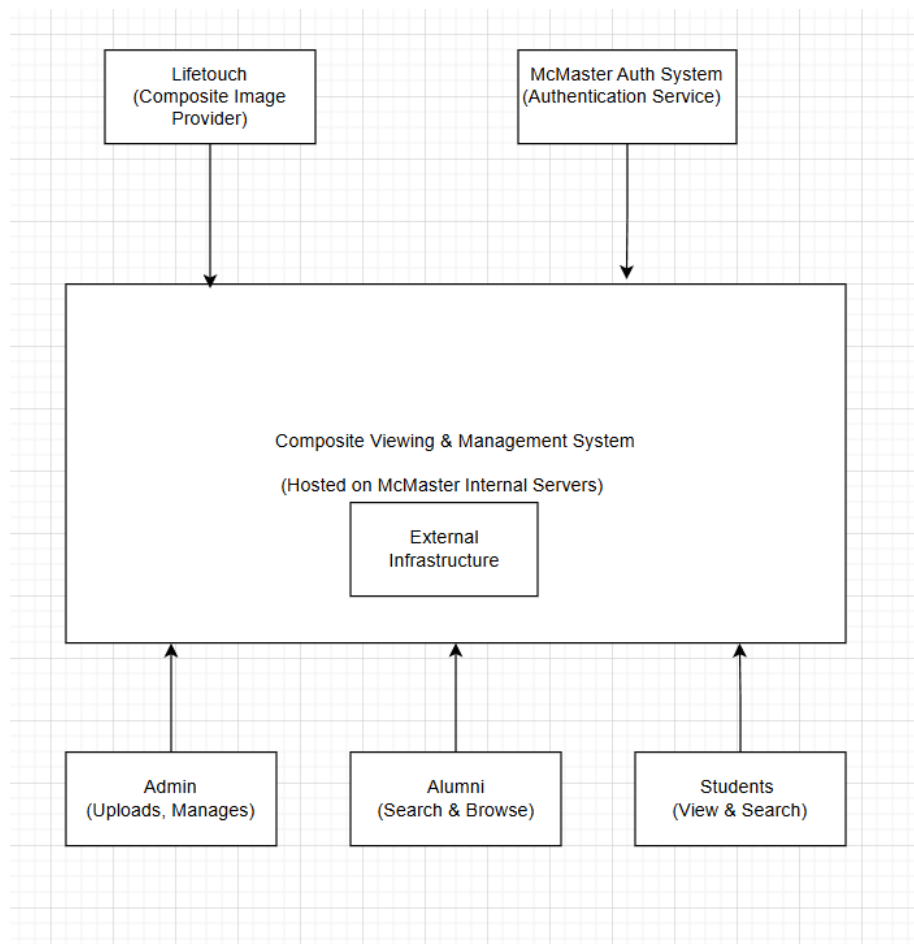


Figure 2: Business Data Model

## 7.2 Data Dictionary

### User

- **UserID (Primary Key):** Unique identifier for each user.
- **FirstName:** First name of the user.
- **LastName:** Last name of the user.
- **Email:** User's email address (used for login and notifications).
- **Role:** User role (e.g., Student, Alumni, Faculty).
- **LastLogin:** Timestamp of the user's last login.

### GraduationComposite

- **CompositeID (Primary Key):** Unique identifier for each graduation composite.
- **Year:** Year of graduation.
- **Program:** Program of study (e.g., Engineering, Arts).
- **Faculty:** Associated faculty (e.g., Faculty of Engineering).
- **CreationDate:** Date the composite was uploaded.

### Student

- **StudentID (Primary Key):** Unique identifier for each student.
- **UserID (Foreign Key):** References User entity.
- **GraduationYear:** Year of graduation.
- **Program:** Program of study.
- **CompositeID (Foreign Key):** References GraduationComposite entity.
- **PhotoURL:** URL link to the student's photo in the composite.

## 8 The Scope of the Product

### 8.1 Product Boundary

#### Internal Functions

- **User Management:**
  - User registration, login, and profile management.
  - Role-based access control (students, alumni, faculty).
  - Secure authentication and password recovery.
- **Graduation Composite Management:**
  - Upload and management of digital graduation composites.
  - Assigning composites to specific years, programs, and faculties.
  - User access to browse and view composites.
- **Search and Interaction:**
  - Search functionality for users to locate composites by year, program, or faculty.
  - Displaying detailed information about a specific composite.
  - Interaction with composites, including zooming in, viewing specific profiles (students in the composite), and related details.

#### External Functions

- **Manual Composite Creation:** The system will not create graduation composites manually (the system manages pre-existing composites).
- **Real-Time Messaging or Chat:** The system does not include real-time communication or chat functionality between users.
- **External Databases Integration:** No direct integration with external student or alumni databases beyond the system's own records.
- **Third-Party Integration:** There is no integration with third-party platforms or social media (e.g., LinkedIn, Facebook) for sharing or viewing composites.

8.2 Product Use Case Table

Table 1: Product Use Case Table

Use Case ID	Use Case Name	Description	Primary Actor(s)	Preconditions	Trigger	Flow of Events (Basic Path)	Postconditions
UC01	User Registration	Allows users to register an account in the system	Student, Alumni, Faculty	The user has not yet registered	User accesses the registration page	1. User enters registration information 2. System validates details 3. Account is created	User is registered and can log in
UC02	User Login	Authenticates users to access the system	Student, Alumni, Faculty	The user is registered and has valid credentials	User submits login credentials	1. User enters email and password 2. System verifies credentials 3. User is logged in	User is logged into the system
UC04	View Graduation Composite	Users can view specific graduation composites	Student, Alumni, Faculty	The user is logged in, and composites are available	User selects a composite to view	1. User selects composite 2. System retrieves and displays the composite 3. User views composite	Composite is displayed for user
UC05	Search Graduation Composite	Users can search for a composite based on filters (year, etc.)	Student, Alumni, Faculty	The user is logged in, and search filters are available	User enters search criteria	1. User enters search criteria 2. System processes the search 3. Matching results are displayed	Relevant composites are listed



### 8.3 Individual Product Use Cases (PUC's)

UC01 → User Registration, can be student or just user

UC02 → User login, can be student or user logging in, in which individual parties have different access

UC03 → View Graduation Composite, the default composite seen, specifically the latest graduation year composite for desired faculty

UC04 → Search Graduation Composite, search feature to be able to search past year composites and students

## 9 Functional Requirements

- **FR-1:** The system shall allow faculty to upload digital graduation composites.
- **FR-2:** The system shall store metadata with each composite, including the year of graduation, program, and associated faculty.
- **FR-3:** The system shall differentiate users by their role (e.g., Student, Alumni, Faculty), granting access to appropriate features based on role.
- **FR-4:** The system shall allow users to view graduation composites, with an option to zoom in on individual profiles.
- **FR-5:** The system shall allow users to search for composites based on the year, program, or faculty.
- **FR-6:** The system shall log user activities, such as logins, composite views, and feedback submissions, for audit purposes.
- **FR-7:** Users should be able to search by name, program, year, etc., with minimal input required.
- **FR-8:** Users should be able to navigate with ease, using intuitive gestures to zoom on class composites, tapping on individual students to view information, etc.
- **FR-9:** The system's search function should match the user's search inputs with relevant class composites, prioritizing exact matches, followed by fuzzy matches.

- **FR-10:** Search filters must display accurate results to the user's selection.
- **FR-11:** In case of network issues, the system should still allow the user to interact and search without an internet connection.
- **FR-17:** The system should be able to recover from any software issues without crashing, displaying an appropriate error message to the user.

## 10 Look and Feel Requirements

### 10.1 Appearance Requirements

- **Display (NFR-1):** The interface should be able to clearly display high-resolution images of class composites, with minimal pixelation.
- **Screen layout (NFR-2):** The interface should use a consistent layout across all pages, including an option to search and filter through the digital composites. The navigation should be easily accessible from any page, allowing new users to immediately interface with the system.

### 10.2 Style Requirements

- **Color Scheme (NFR-3):** A color palette should be selected to avoid distraction from viewing the digital composites.
- **Font Choice (NFR-4):** A legible, sans-serif font should be used, that can also scale with varying screen sizes and accessibility needs.
- **Iconography (NFR-5):** Intuitive icons should be used, to make it universally recognized and easy to identify.
- **Consistency (NFR-6):** A uniform style for buttons, margins, padding, etc., should be used across all pages and screens.

## 11 Usability and Humanity Requirements

### 11.1 Ease of Use Requirements

- **Touch Sensitivity (NFR-7):** The interface must be responsive and accurate to touch gestures (tapping, scrolling, swiping, etc.)
- **Search Functionality (FR-7):** Users should be able to search by name, program, year, etc., with minimal input required.
- **Navigation (FR-8):** Users should be able to navigate with ease, using intuitive gestures to zoom on class composites, tapping on individual students to view information, etc.

### 11.2 Personalization and Internationalization Requirements

N/A

### 11.3 Learning Requirements

- **Minimal Learning Curve (NFR-8):** The interface should be easy and straightforward to use, allowing users to immediately interact with the system with minimal guidance.
- **On-Screen Guidance (NFR-9):** Contextual tooltips or guidance should be displayed when users interact with the interface for the first time.

### 11.4 Understandability and Politeness Requirements

- **Clarity (NFR-10):** All instructions should be clear and concise to all users who may interact with the system.
- **Politeness (NFR-11):** The system should provide polite and informative error messages, in case of failed user interactions, or technical issues caused by the system.

### 11.5 Accessibility Requirements

- **Accessibility Features (NFR-12):** The system should offer support for a more accessible interface, with large, touch-friendly buttons and text.

## 12 Performance Requirements

### 12.1 Speed and Latency Requirements

- **Search Speed (NFR-13):** The interface should update with search results within 1-2 seconds of user input.
- **Page Load Time (NFR-14):** Any new screen should load in under 2 seconds.
- **Smooth Animations (NFR-15):** Any transitions/animations used should be smooth, without stuttering or lag.

### 12.2 Safety-Critical Requirements

- **Data Security (NFR-16):** All personal information related to students is protected from unauthorized access.
- **Error Handling (FR-17):** The system should be able to recover from any software issues without crashing, displaying an appropriate error message to the user.

### 12.3 Precision or Accuracy Requirements

- **Search Accuracy (FR-9):** The system's search function should match the user's search inputs with relevant class composites, prioritizing exact matches, followed by fuzzy matches.
- **Filter Accuracy (FR-10):** Search filters must display accurate results to the user's selection.

### 12.4

sectionRobustness or Fault-Tolerance Requirements

- **Offline Capability (FR-11):** In case of network issues, the system should still allow the user to interact and search without an internet connection.

### 12.5 Capacity Requirements

- **Concurrent Users (NFR-17):** The interface should be able to handle multiple users interacting with different touch screen displays without any substantial decrease in performance.

## 12.6 Scalability or Extensibility Requirements

- **Scalability (NFR-18):** The system should easily scale with each additional year of graduating class composites over time, without a significant decrease in performance. The database of composites should automatically sync with any new uploads of composites.
- **Extensibility (NFR-19):** The system should be developed to allow for new features to be added in the future.

## 12.7 Longevity Requirements

- **Maintenance (NFR-20):** The system should be designed to require minimal maintenance while scaling with more digital composites over time.

# 13 Operational and Environmental Requirements

## 13.1 Expected Physical Environment

- **Indoor Placement (NFR-21):** The interface will be placed indoors in buildings across McMaster University campus. The system must withstand its durability from frequent users interacting with it.
- **Lighting (NFR-22):** The system's screen should be visible and functional in varying light conditions.

## 13.2 Wider Environment Requirements

- **Building Infrastructure (NFR-23):** The system should integrate seamlessly into existing campus buildings, allowing installation to only require power and network access.
- **Security (NFR-24):** The physical device interface must be secure from tampering from an unknown third-party.

## 13.3 Requirements for Interfacing with Adjacent Systems

### External Data Integration

- **FR-12:** The system shall integrate with McMaster University’s internal databases to retrieve graduation records and available digital composite images.
- **FR-13:** The system shall support secure connections to external providers (e.g., LifeTouch) to retrieve high-resolution composite photos of alumni, in compliance with institutional data agreements.

### Legacy Composite Image Parsing

- **FR-14:** The system shall allow the uploading and automated parsing of scanned legacy composite images, including extraction of names and image segmentation, without requiring manual input.
- **NFR-25:** The system shall support scalability to handle large datasets spanning multiple years of graduating classes.

### Data Communication & Security

- **NFR-26:** All data exchanged between the GradSight system and external systems shall be encrypted and transmitted over secure, authenticated API connections.
- **P-4:** The average response time for data retrieval operations from external systems shall not exceed 3 seconds under normal load.
- **S-3:** The system shall validate all retrieved composite data before incorporating it into the internal database, ensuring integrity and format consistency.

## 13.4 Productization Requirements

The system must be scalable and operable across different hardware environments, including desktop and other interfaces. The productization strategy will focus on providing user-friendly setup guides that can help deploy the product easily across different faculties. Also, the solution must cater to non-technical users, ensuring simple management post-deployment. When it comes to the performance metrics, it should be able to support up to all of the composite images without significant degradation and operate efficiently on both touchscreen and desktop setups.

- **PR-1:** The system shall support deployment on both touchscreen-based kiosks and standard desktop workstations without requiring separate codebases.
- **PR-2:** The system shall include a user-friendly setup guide tailored for non-technical users, enabling deployment by faculty or IT support with minimal training.
- **PR-3:** The system shall allow administrative users to manage and update content (e.g., composite uploads, metadata edits) via a simplified interface that does not require coding or technical knowledge.
- **PR-4:** The system shall be capable of storing and displaying all composite images across years (target: 50+ years of data) with no more than a 10% degradation in performance benchmarks such as load time and search latency.
- **PR-5:** The system shall operate efficiently on commonly available faculty hardware setups (e.g., Raspberry Pi, Intel NUC, standard PCs) without requiring specialized hardware or cloud infrastructure.

## 13.5 Release Requirements

The project will follow an agile development approach, with iterative updates and continuous feedback throughout the development phase. However, the release strategy will adopt a waterfall model, meaning the system will only be released once all development and testing stages are complete. This ensures that the system is stable, rigorously tested, and fully developed before being deployed. Moreover, no interim public releases will be made, and the final deployment will occur only when the system reaches a production-ready state, ensuring minimal post-release issues.

- **RR-1:** The system shall be released only after all planned development tasks, integration, and testing phases have been completed and approved.
- **RR-2:** No interim public or campus-wide releases shall be made prior to the final release, except for internal team testing and demonstrations during development milestones.
- **RR-3:** The final release version shall meet all functional and nonfunctional requirements outlined in this document, and must pass system-wide verification and validation tests.

- **RR-4:** The release plan shall include a rollback procedure to restore the previous system state in case of critical post-release issues.
- **RR-5:** A final production-ready release shall be packaged with full documentation (e.g., setup instructions, user guides, maintenance procedures) and delivered to McMaster IT Services for deployment.

## 14 Maintainability and Support Requirements

### 14.1 Maintenance Requirements

The system's codebase will be modular, designed to simplify future maintenance by developers who may not have been involved in the original development. All technical documentation, including system architecture, coding standards, and troubleshooting steps, will be housed in a GitHub repository. Also, the end-user documentation will be provided to assist users during the initial deployment phase, along with an advisory system to help address early-stage issues.

- **SUP-1:** The system's codebase shall follow a modular structure to allow isolated updates and simplify maintenance tasks.
- **SUP-2:** Technical documentation, including system architecture diagrams, API descriptions, coding standards, and setup instructions, shall be maintained in a GitHub repository accessible to McMaster IT Services.
- **SUP-3:** End-user documentation (e.g., deployment guide, feature descriptions, admin interface usage) shall be delivered alongside the final release package.
- **SUP-4:** An advisory guide shall be provided to help address common setup and troubleshooting issues, particularly during early-stage deployment and usage.
- **SUP-5:** The system shall be maintainable by developers who did not participate in the original development, without requiring significant re-familiarization time.

### 14.2 Supportability Requirements

For long-term support, the system will maintain detailed logs that allow administrators to diagnose issues and track user interactions. Support will be delivered through



an FAQ page and online documentation, reducing the need for hands-on assistance. When it comes to error handling, the messages must be clear and actionable, enabling users to quickly identify and address points of failure without requiring external support. Logs will be retained ensuring compliance and helping with future diagnostics analysis if needed.

- **SUP-6:** The system shall maintain detailed activity logs capturing key events, system errors, and user interactions to aid in diagnostics and compliance audits.
- **SUP-7:** All log data shall be securely stored and retained for a minimum of 12 months to support future debugging, analysis, or compliance requirements.
- **SUP-8:** The system shall provide an FAQ page accessible from the main interface to address common user issues and questions.
- **SUP-9:** Online user documentation shall be hosted with the system and updated regularly to reflect system changes and new features.
- **SUP-10:** All system-generated error messages shall be clear, concise, and actionable to help users resolve issues without requiring technical support.

### 14.3 Adaptability Requirements

The system must be adaptable to both hardware and software changes, supporting different display types and resolutions, from desktop monitors to large touchscreen displays. The system must also handle composite images with varying layouts and qualities, ensuring flexibility for future-proofing. The system's performance should remain consistent across different hardware setups, with minimal configuration required when adapting to new environments. Lastly, it should scale to handle future growth in data volume while maintaining minimal response times to ensure user satisfaction.

- **ADP-1:** The system shall support various display types and screen resolutions, ranging from desktop monitors to large touchscreen interfaces, without requiring code changes.
- **ADP-2:** The system shall correctly render composite images of different layouts, formats, and resolutions, ensuring consistent functionality regardless of input source quality.

- **ADP-3:** The system shall automatically detect and adjust to different hardware configurations during installation, minimizing manual configuration.
- **ADP-4:** The system shall maintain consistent performance across different supported hardware environments (e.g., Raspberry Pi, mini-PC, desktop), with no noticeable degradation in functionality or responsiveness.
- **ADP-5:** The system shall be scalable to accommodate increases in the number of composite images and user interactions without exceeding a 2-second average response time under normal load.

## 15 Security Requirements

### 15.1 Access Requirements

Role-based access controls should be implemented so that different levels of access can be granted (e.g., faculty staff vs. admin users). Moreover, all data transmissions must be encrypted to prevent unauthorized access during communication with external systems.

- **SEC-1:** The system shall implement role-based access control (RBAC) to restrict access based on user roles, such as faculty staff and administrators.
- **SEC-2:** The system shall ensure that all data transmissions—both incoming and outgoing—are encrypted using secure protocols (e.g., HTTPS/TLS) to prevent unauthorized access during communication with external systems.

### 15.2 Integrity Requirements

Data integrity is crucial, particularly when processing historical composite images and student data, so the system must validate all input data to ensure that no corrupted or malicious data enters the system. Moreover, version control of images and data must be implemented to track changes and revert to previous states if necessary.

- **INT-1:** The system shall validate all input data—including uploaded composite images and student metadata—to ensure data is not corrupted or malicious before it is accepted into the system.

- **INT-2:** The system shall implement version control for all stored images and associated metadata, enabling administrators to track changes and revert to previous versions when necessary.

### 15.3 Privacy Requirements

The system must comply with data privacy regulations, such as GDPR, ensuring that personal student information (names, photos) is handled securely. This means that all personal data must be encrypted both at rest and in transit. Any breach or unauthorized access should be logged and reported to the support staff for appropriate measures.

In addition, data minimization principles should be followed — only the necessary personal information required for system functionality should be collected and stored. The system must include user consent mechanisms where applicable, allowing users to give explicit consent before their data is used. A clear privacy policy should be made available within the system to inform users about how their data is processed, stored, and shared.

Finally, anonymization techniques should be considered for non-essential personal data, ensuring that sensitive information is not directly linked to an identifiable person unless strictly necessary.

- **PRIV-1:** The system shall comply with relevant data privacy regulations, including GDPR, ensuring all personal student data (e.g., names, photos) is securely handled.
- **PRIV-2:** The system shall encrypt all personal data both at rest and in transit.
- **PRIV-3:** The system shall log and report any unauthorized access or data breaches to the designated support staff.
- **PRIV-4:** The system shall follow data minimization principles, collecting and storing only the personal information necessary for its functionality.
- **PRIV-5:** The system shall implement user consent mechanisms, allowing users to give explicit consent before their personal data is used.
- **PRIV-6:** The system shall include a clear, accessible privacy policy that outlines how data is processed, stored, and shared.

- **PRIV-7:** The system shall apply anonymization techniques to non-essential personal data, ensuring that such data cannot be traced back to individuals unless strictly required.

## 15.4 Audit Requirements

The system must maintain detailed audit logs, tracking all system interactions and access events in a tamper-proof format and accessible only by system administrators. To ensure security and compliance, these audit logs must be stored in a secure, encrypted format to prevent unauthorized viewing or manipulation of the logs. The system should also implement integrity checks to detect any potential tampering, with alerts triggered in the event of suspicious activity. Audit logs must be organized in a way that facilitates easy retrieval and review, allowing system administrators to quickly investigate any access patterns, system anomalies, or suspected breaches.

- **AUD-1:** The system shall maintain detailed audit logs that track all system interactions and access events.
- **AUD-2:** Audit logs shall be stored in a tamper-proof format, accessible only by authorized system administrators.
- **AUD-3:** Audit logs shall be encrypted at rest to prevent unauthorized viewing or manipulation.
- **AUD-4:** The system shall implement integrity checks on audit logs to detect potential tampering.
- **AUD-5:** The system shall trigger alerts in the event of suspicious or unauthorized access patterns.
- **AUD-6:** Audit logs shall be organized to facilitate easy retrieval and review by system administrators.
- **AUD-7:** The system shall enable administrators to investigate anomalies, access history, and suspected breaches using the audit logs.

## 15.5 Immunity Requirements

The system must be designed to resist common cyber threats, including SQL injection, cross-site scripting, and denial-of-service attacks. To achieve this, the system

will leverage GitHub’s security monitoring features, which automatically detect vulnerabilities in third-party packages and dependencies. This ensures that any critical vulnerabilities in the libraries or frameworks used are promptly identified and provide automated security patches and updates. Doing so will minimize the window of exposure to known vulnerabilities and harm.

- **SEC-1:** The system shall be resilient to common cyber threats, including but not limited to SQL injection, cross-site scripting (XSS), and denial-of-service (DoS) attacks.
- **SEC-2:** The system shall sanitize all user inputs to prevent injection-based attacks.
- **SEC-3:** The system shall implement rate limiting and input validation to mitigate the impact of DoS attacks.
- **SEC-4:** The system shall utilize GitHub’s security monitoring features to automatically detect known vulnerabilities in third-party packages and dependencies.
- **SEC-5:** The system shall apply automated updates or notify administrators when critical security patches are available for external libraries.
- **SEC-6:** The system shall log and alert administrators if repeated attack patterns (e.g., multiple failed login attempts or malformed requests) are detected.

## 16 Cultural Requirements

The system will only support searching for alumni in English, which simplifies the interface and avoids the complexity of supporting multiple languages. However, it must ensure that all names, including those with special characters or accents, are accurately processed and displayed in the English-language interface. This will ensure that users from diverse backgrounds can still be represented correctly, even though the primary language of interaction is English.

- **CUL-1:** The system shall support searching for alumni exclusively in English, with no multilingual functionality provided at launch.
- **CUL-2:** The system shall correctly process and display names containing special characters, diacritics, or accents in the English-language interface.

- **CUL-3:** The system shall store all name data in a Unicode-compliant format to preserve international character accuracy and ensure representation of diverse user backgrounds.

## 17 Compliance Requirements

### 17.1 Legal Requirements

The system must comply with all relevant intellectual property laws, particularly when using third-party data such as images from LifeTouch. The system should also include mechanisms for ensuring that licenses (such as the Apache 2.0 license) are adhered to and any legal changes that could affect the system's operation should be documented.

- **LGL-1:** The system shall comply with all applicable intellectual property laws, including those governing the use of third-party data and images provided by LifeTouch.
- **LGL-2:** The system shall ensure that all third-party software and libraries used are appropriately licensed (e.g., Apache 2.0), and that license obligations are documented and followed.
- **LGL-3:** The system shall maintain a record of legal and licensing requirements, including versioned documentation of any changes affecting system compliance.

### 17.2 Standards Compliance Requirements

The system must adhere to ISO/IEC 25010 standards for software quality and GDPR for data privacy. Also, all compliance-related features (e.g., encryption, data retention) should be automatically tested during each major release to ensure ongoing conformity.

- **LGL-4:** The system shall conform to the ISO/IEC 25010 standard for software quality, including characteristics such as reliability, maintainability, and usability.
- **LGL-5:** The system shall comply with the General Data Protection Regulation (GDPR) for handling and storing personal data, including encryption, access control, and data minimization.

- **LGL-6:** All compliance-related functionality (e.g., encryption, data retention, access auditing) shall be included in the automated test suite and executed during each major release cycle to verify continued standards compliance.

## 18 Open Issues

The system currently faces potential issues with hardware as the final device to host this product isn't finalized yet. Additionally, OCR performance with low-quality images may require further calibration, as preliminary tests indicate a higher error rate than anticipated. Moreover, the individualized images from LifeTouch are also not confirmed, so the expectations may change if they decide not to give those.

## 19 Off-the-Shelf Solutions

### 19.1 Ready-Made Products

The system leverages existing solutions such as GitHub for version control and CI/CD pipeline. This will allow for pre-commit and post-merge testing to ensure robustness. Moreover, Optical character recognition (OCR) will be handled by an existing model (yet to be determined) to save time and resources on custom development.

### 19.2 Reusable Components

The system should be designed with reusability in mind, utilizing pre-trained OCR models for composite image parsing, reusable front-end components, and a modular back-end architecture. These components must be adaptable for future projects or system upgrades, so it'll be easier if changes are needed.

### 19.3 Products That Can Be Copied

There are currently no products that can be directly copied. However, we can pull some ideas used in the digital composite wall at UBC.

## **20 New Problems**

### **20.1 Effects on the Current Environment**

The installation of the digital composite in John Hodgins Engineering Building and into the university's IT environment may introduce new network loads, and potential crowding in the building during peak usage (e.g., graduation season, or alumni events).

### **20.2 Effects on the Installed Systems**

Effects on the installed systems include requiring more space for existing databases, introduction of potential new security protocols to meet with university safety and privacy concerns, and separated processing to ensure functionality without affecting other ongoing services.

### **20.3 Potential User Problems**

Some potential problems users may encounter would be difficulties with the touch-screen interface or with navigating the system. An easy to use interface, and tooltips within the app will be provided to ensure accessibility and ease of use for non-technical users.

### **20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

Limitations include the existing security protocols and server configurations that may restrict how the app communicates with McMaster's private servers.

### **20.5 Follow-Up Problems**

Maintenance issues could arise after the installation of the digital composite. In particular these would include updates to the database, server configurations, and security protocols. A team will be required to regularly check in on the system to update and troubleshoot to prevent downtime and ensure smooth operation.



## 21 Tasks

### 21.1 Project Planning

The project will primarily follow the plan described in the Capstone Outline:

- Team Formed, Project Selected [Sept 16]
- Problem Statement, POC Plan, Development Plan [Sept 23]
- Requirements Document Revision 0 [Oct 9]
- Hazard Analysis 0 [Oct 23]
- V&V Plan Revision 0 [Nov 1]
- Proof of Concept Demonstration [Nov 11–22]
- Design Document Revision 0 [Jan 15]
- Revision 0 Demonstration [Feb 3–14]
- V&V Report Revision 0 [Mar 7]
- Final Demonstration (Revision 1) [Mar 24–30]
- EXPO Demonstration [April TBD]
- Final Documentation (Revision 1) [Apr 2]

However, in between these deliverables we will follow the typical software development lifecycle. This starts with requirements gathering, design, development, testing, and deployment. Feedback will be gathered across all stages to ensure the app meets usability and performance requirements.

## 22 Planning of the Development Phases

### Phase 1: Requirements Gathering and System Design

Goals:

- Finalize user personas and stakeholder priorities

- Create system architecture diagrams and the context diagram
- Identify high-level technical constraints and design assumptions

**Requirements Addressed:**

- FR-1 to FR-3 (Core functionality outline)
- NFR-1 to NFR-4 (Visual design and layout expectations)
- CUL-1 to CUL-3 (Cultural assumptions and name display handling)
- LGL-1, LGL-2 (Legal use of third-party assets and licenses)

## **Phase 2: Core System Development (Frontend and Backend)**

**Goals:**

- Build the frontend UI and admin panel
- Set up the backend database and APIs
- Integrate machine learning module for name detection/parsing

**Requirements Addressed:**

- FR-4 to FR-14 (Search, filtering, OCR, integration with external data)
- NFR-5 to NFR-15 (Touch response, animation smoothness, accessibility, etc.)
- ADP-1 to ADP-4 (Adaptability across devices and resolution handling)
- PR-1 to PR-3 (Productization across faculties)

## **Phase 3: Security Compliance & System Hardening**

**Dates:** Jan 15 – Feb 3

**Goals:**

- Implement all encryption and role-based access controls
- Run compliance checks (GDPR, ISO/IEC 25010)
- Build out audit logging and testing infrastructure

**Requirements Addressed:**

- SEC-1 to SEC-6 (Security and immunity)
- INT-1, INT-2 (Integrity validation and rollback)
- PRIV-1 to PRIV-7 (Privacy compliance)
- AUD-1 to AUD-7 (Auditing infrastructure)
- LGL-3 to LGL-6 (Compliance verification)

**Phase 4: User Testing & Feedback****Goals:**

- Conduct usability testing on hardware
- Collect metrics on user interaction and satisfaction
- Incorporate accessibility and UI feedback

**Requirements Addressed:**

- NFR-16 to NFR-22 (Security, accessibility, responsiveness)
- SUP-6 to SUP-9 (FAQ, documentation)
- FR-7 to FR-11 (User interactions, search, filters, offline use)

**Phase 5: Final Deployment and Post-Deployment Support****Goals:**

- Final release to McMaster IT
- Deliver full documentation
- Conduct final performance and reliability checks
- Begin support and monitoring

**Requirements Addressed:**

- RR-1 to RR-5 (Release verification and rollback support)

- SUP-1 to SUP-5 (Maintenance readiness)
- ADP-5 (Performance scaling under user load)
- PR-4, PR-5 (Final scaling and performance assurance)

## **23 Migration to the New Product**

### **Requirements for Migration to the New Product**

The new product will not be replacing any existing systems but will be an addition to the university. The only data that will need to be migrated are the digital composites provided 2020–2024, but otherwise data will need to be uploaded and maintained by administrators yearly.

### **Data That Has to be Modified or Translated for the New System**

Data from older, physical composites need to be scanned and converted to a digital format. Additionally, the format will be parsed through by machine learning models to retrieve student names.

### **Costs**

The primary cost for this project would be the hardware. This includes the large touchscreen for the physical interface and its accessories. Secondary costs would relate to server usage and ensuring compliance with security and privacy standards.

## **24 User Documentation and Training**

### **User Documentation Requirements**

Comprehensive documentation will be provided for both end-users and the administrators responsible for maintaining and updating the system. Documentation will include instructions for using the touch screen interface and the search functionality. For the administrators, the documentation will include how the complete system functions, how to update it, and how to troubleshoot.

## **Training Requirements**

Administrators will require brief training on how to convert composites, upload them, manage user analytics and address privacy concerns. Training will also include (specifically for developers) how to troubleshoot basic issues such as server connection and app functionality.

## **25    Waiting Room**

Additional features like integration with alumni services and expanding student details are possibilities for future iterations of the system but are not part of the current scope.

## **26    Ideas for Solution**

For future improvements, the machine learning models could be further enhanced to parse more complex composites or provide better analytics. Integration with alumni networks or graduation services could provide a more personalized experience for users.

## 27 Traceability Matrix

Table 2: Table X: Traceability Matrix

Requirement ID	System Component / Feature	Requirement ID	System Component / Feature
FR-1	Login System	NFR-15	Animations
FR-2	Login System	NFR-16	Database
FR-3	User Interface	NFR-17	Concurrency Engine
FR-4	UI Search	NFR-18	Database Engine
FR-5	Search Filters	NFR-19	Maintenance Tools
FR-6	Image Viewer	NFR-20	Touchscreen
FR-7	UI Search	NFR-21	Display
FR-8	Image Viewer	NFR-22	Infrastructure
FR-9	Search Engine	NFR-23	Tamper Protection
FR-10	Search Filters	NFR-24	Data Sync
FR-11	Offline Mode	NFR-25	API Layer
FR-12	Database	NFR-26	API Response Time
FR-13	Image Retrieval	P-4	Data Validation
FR-14	OCR Engine	S-3	Cross-Hardware Support
FR-17	Error Handler	PR-1	Setup Docs
NFR-1	Composite Display	PR-2	Admin Panel
NFR-2	Composite Layout	PR-3	Data Capacity
NFR-3	UI Theme	PR-4	Hardware Compatibility
NFR-4	Fonts	PR-5	Release Plan
NFR-5	Icons	RR-1	Internal Testing
NFR-6	Layout System	RR-2	V&V Test Suite
NFR-7	Touch Input	RR-3	Rollback Plan
NFR-8	UI Simplicity	RR-4	Release Documentation
NFR-9	Tooltips	RR-5	Modular Code
NFR-10	Text Content	SUP-1	Documentation
NFR-11	Error Messaging	SUP-2	End User Docs
NFR-12	Accessibility	SUP-3	Troubleshooting
NFR-13	Search Engine	SUP-4	Developer Friendliness
NFR-14	Page Renderer	SUP-5	Logging System

## Appendix — Reflection

### Knowledge and Skills Required

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning.

**What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project?**

Examples of possible knowledge to acquire include domain-specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

- **Hammad** — One skill that our team will collectively need to acquire to successfully complete this capstone project will be learning privacy and security protocols and how to implement them.
- **Willie** — To successfully complete this capstone project, our team will need to know how to develop with frontend frameworks, one that is specific to handling touch inputs from a user. We will need to understand how to both program and deploy our interface solution to a display, running our application locally to the device.
- **Zahin** — Our team will need to learn skills in data integration, which involves connecting the application up to the university's existing systems or outside data sources, such as Lifetouch. This knowledge is going to be important in ensuring ease of movement of data and updating of a system with the latest information.
- **Henushan** — We would need to learn how to deploy code using frameworks and tools like Jenkins and Helios. When we are working with a user interactive screen, a scalable server would need to be in place.
- **Wajdan** — A skill that our team needs to learn is how to make and leverage reusable components. By creating components that can be used across multiple projects or areas of the same project, we can save time and reduce redundant work. It'll help us maintain consistency in both design and functionality, making sure that our work stays efficient and scalable. Also, learning how to do this means understanding how to break down our code or designs

into smaller, modular pieces that can be easily adapted and reused without constant rewriting.

**For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?**

- **Hammad** — For privacy and security protocols, one approach to this would be watching videos and reading documents on how these are implemented and used in projects. The other approach would be to work with the developers currently at McMaster and LifeTouch to understand how the protocols used on their systems currently work. I will choose the first approach because I can learn more in depth and at my own pace to understand. In addition, I can reach out to the developers afterwards for further information.
- **Willie** — To acquire the skill of frontend development, one approach would be to research any of the widely available online resources on the topic. Where tutorial videos and documentation is readily available to learn from, I will be able to understand how to develop and deploy our solution to our digital interface. Another approach would be to learn by practice, attempting to develop smaller-scale applications until I am more comfortable with the knowledge, so I can apply it towards the success of our capstone project.
- **Zahin** — To acquire the skills in backend development and database management, one approach would be online classes or tutorials focused on backend programming and designing databases, which will give a good foundation. The other way would be talking to the senior developers at McMaster who deal with the university's data systems for practical insights and best practices. We will adopt the former approach because it gives us an ample opportunity to understand the concepts through structured learning before applying them to the project.
- **Henushan** — To deepen our understanding in deployment and dev-ops tools, we would need to review and go through documentation and even watch tutorials on infra and setup. From project to project, it will be different but the underlying steps should be similar which we can work and improve on. Additionally, working on setting up personal projects using these tools would prove to be a useful aid. With more practice and familiarity with these tools, our deployment practices will inevitably improve.



- **Wajdan** — To properly understand how to build reusable components, we would need to grasp a few key concepts. First, we need to understand modularity—how to break down complex systems into smaller, independent units that can function in isolation but also work seamlessly when combined. We also need to master separation of concerns, ensuring each component handles a single responsibility to make it easier to maintain and reuse. The team will use both of these approaches in order to get more practice and get hands-on experience when dealing with reusable components.