Faculty of Engineering and Applied Science

ENGR 4940U Capstone Systems Design for ECSE I

Design and Development of a Virtual Reality System to support Reminiscence Therapy for Patients with Dementia

R2: Concept Generation, Conceptual Design and Prototype

Team Members

Abdurrahman Ansari, 100585644

Dhanushga Lionel, 100616831

Mohammed Hameeduddin, 100587893

Mingwei Zhang, 100519827

Faculty Advisor: Dr. Ramiro Liscano

Capstone Coordinator (Fall 2019): Dr. Q. Mahmoud

Table of Contents

List of Tables

List of Figures

1. Concept Generation and Analysis

Our main project function is to wirelessly stream media content to a display device from a media server using a webpage portal. The main components in our project are the user interface, database query, and stream service. Our main focus is working on a system that can wirelessly connect a computer to a display device.

There are two ways to solve this problem:

1. Create media streaming device
2. Explore existing off the shelf solutions.

Creating a new media streaming device using computer hardware such as a Raspberry Pi, would serve as a solution. A factor we have to also consider in longevity, our solution as a whole needs to be able to perform without failure. If a failure were to occur, the user would need to understand how to diagnose or even troubleshoot the program. If we continued with creating our own streaming device then the troubleshooting and maintenance would be problematic in the future.

The advantages of using an off the shelf solution are:

1. Existing technology is already developed by big companies and tested for failure
2. For popular solutions there are many guides and tutorials available online
3. Products come with manufacturer's warranty and support incase of failure.

After discussing and weighing our different options we have decided to go with an off the shelf solution for our media streaming component. We will be integrating the functionality of the Google Chromecast with the web portal that will be used to query the database. The web portal will be created by us and will be projected to whichever display device the user selects.

The figure below details how our project components will interact with each other.

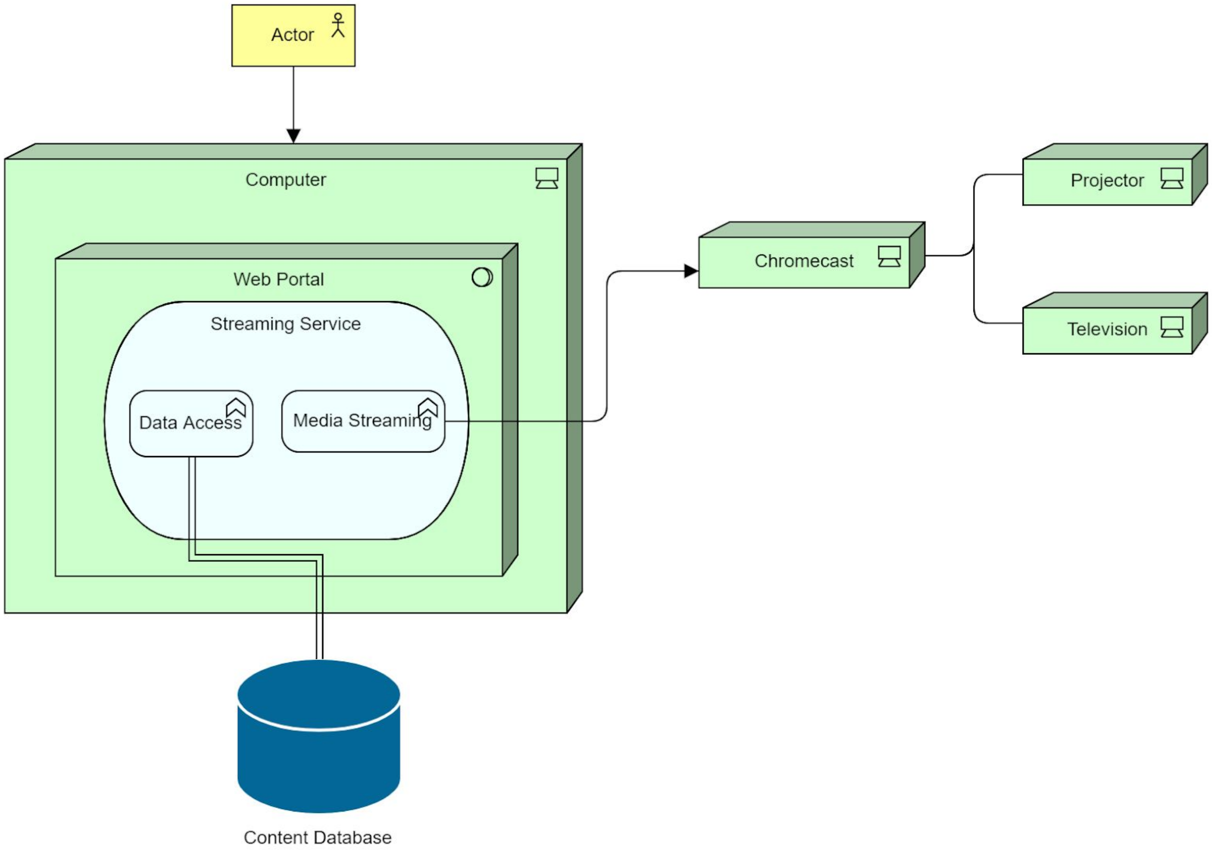
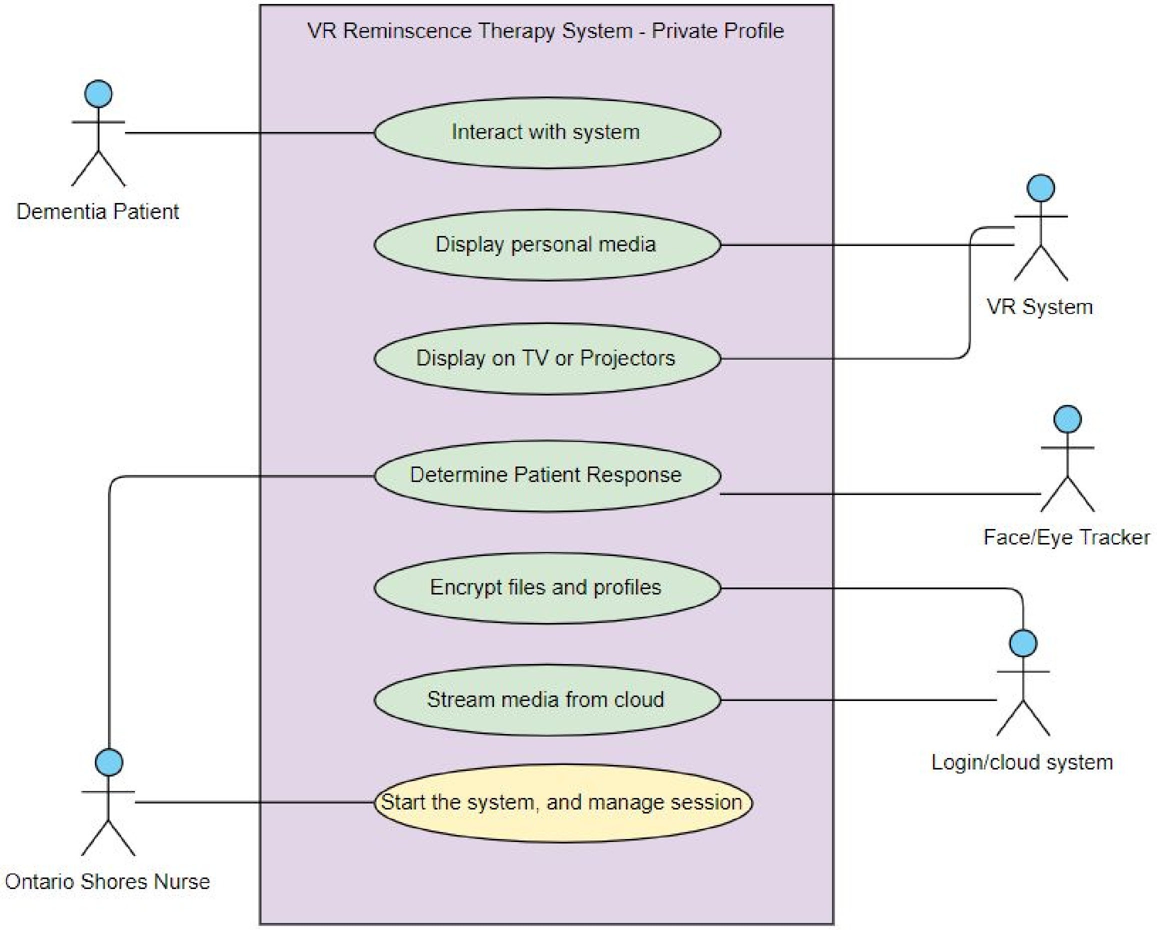


Figure 1.1 Relational Model

1b. ADD Design Process


VR Reminscence Therapy System - Private Profile
Dementia Patient
Interact with system
Display personal media
VR System
Display on TV or Projectors
Determine Patient Response
Face/Eye Tracker
Encrypt files and profiles
Stream media from cloud
Login/cloud system
Start the system, and manage session
Ontario Shores Nurse

Use Case	Description
UC-1	Users, such as nurses, administrators and caregivers of the patients need to interact with the system
UC-2	The VR system needs to display the general profile
UC-3	VR system should push content onto a projected surface or tv using a chromecast or similar device
UC-4	Ontario Shores Personnel, whether it be nurses, or administrators, needs to be able to start the system, and manage session
UC-5	VR System needs to display personal profiles

	of each individual patient
UC-6	User needs to login to the system, using 2-factor authentication for extra protection
UC-7	Media files need to be compressed, and encrypted
UC-8	Profiles should be able to be streamed from a cloud database

Quality Attributes

ID	**Quality Attribute**	**Scenario**	**Associated Use Case**	**Priority**
QA-1	Performance	● Multiple users needs to be able to access system, without any data loss ● System should be scanned periodically for errors so as to not cause shut-down	UC-1, UC-6	H,H
QA-2	Modifiability	● New profiles should be able to be accessed, without system crashing	UC-1, UC-6	M,M
QA-3	Availability	● In case of failure, system can shut down and restart	All	H,H
QA-4	Usability	● Profile user can access should be displayed. System should be fast and capable of making quick changes ● Easy interface for everyone to be able to use	UC-4	H,M
QA-5	Security	● If changes are made to the	All	H,H

		system, it is possible to know which user made the change ● If there is any unauthorized access, system should know ● If any user is logging in, 2-factor authentication, needs to be implemented ● If any user uploads media files, files should be safe from security breaches ● System should be safe from critical junction of merging both projects		

Constraints

ID	**Constraints**
CON-1	Multiple users should be able to login, and use system at the same time without failure
CON-2	Relational database needs to be used
CON-3	Changes made should be stored for data recovery purposes
CON-4	Nurse/Administrator should be able to access all profiles, and can stream both public and private profiles
CON-5	Private Wi-fi network, so chromecast cannot be used by unauthorized persons
CON-6	Systems shall not allow users to change information maintained by secondary systems

Architectural Concerns

ID	**Concern**
CRN-1	Establish overall system structure
CRN-2	Leverage team vast programming knowledge, and various programming languages
CRN-3	Allocate work to members of development team

Iteration 1

ADD Step 1: Review Inputs

<table>
<tr><th>Category</th><th>Details</th></tr>
<tr><td>Design Process</td><td>This is a greenfield system from a novel domain. The purpose is to produce a sufficiently detailed design to support the construction of the system.</td></tr>
<tr><td>Primary Functional Requirements</td><td>From the uses cases presented from the first iteration, the primary uses cases are
●</td></tr>
<tr><td>Quality Attribute Scenarios</td><td>
<table>
<tr><td>Scenario ID</td><td>Importance to Stakeholders</td><td>Difficulty of implementation</td></tr>
<tr><td>QA-1</td><td>High</td><td>High</td></tr>
<tr><td>QA-2</td><td>Medium</td><td>Medium</td></tr>
<tr><td>QA-3</td><td>High</td><td>High</td></tr>
<tr><td>QA-4</td><td>High</td><td>Medium</td></tr>
<tr><td>QA-5</td><td>High</td><td>High</td></tr>
</table>
</td></tr>
</table>

	Only QA-1, QA-3, QA-4, QA-4, QA-5 were selected as drivers
Constraints	All constraints discussed in first deliverable are included as drivers
Architectural Concerns	All the architectural concerns discussed in first deliverable are included as drivers

Iteration 1:

This first iteration of the system is to establish an overall system structure

Step 2: Establish Iteration Goal by Selecting Drivers

This is the first iteration in the design of the system, so the goal of this iteration is to *establish an overall system structure*.

Although this iteration is driven by a general architectural concern, the developer must keep in mind all of the drivers that may influence the general structure of the system. In particular the developer must be mindful of the following:

- QA-1: Performance
- QA-3: Availability
- QA-4: Usability
- QA-5: Security
- CON-1: Multiple users without server failure
- CON-2: Relational database

Step 3: Choose one or more elements of the system to refine

This is a greenfield development effort, so in this case, the element to refine is the entire system, which is shown in Figure 1. In this case, refinement is performed through decomposition.

Figure 1: Context diagram for system

Step 4: Choose one or more design concepts that satisfy the selected drivers

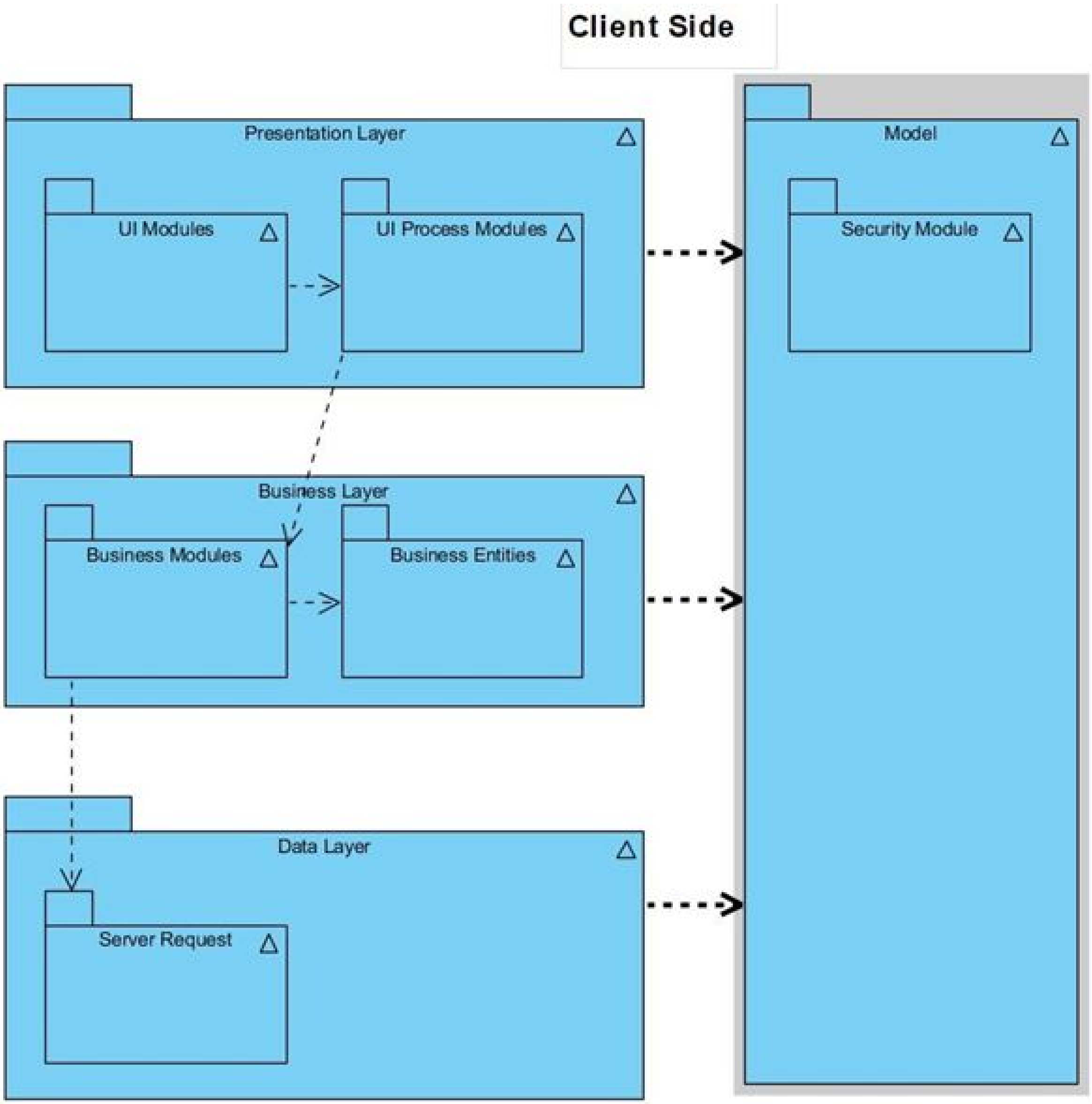
Design Decisions and Location	Rationale
Logically structure the client part of the system using a web application reference architecture	This type of reference architecture must be accessed from a web browser, such as examples of Netflix. This is the best option to implement the client part of the system because it has a User Interface, Business Logic and Data Access, which is what we need for this project. Discarded Alternatives: ● Mobile Application: This type of architecture is only geared towards development of applications that are deployed in handheld devices. This alternative was discarded because this type of device was not considered for the System. ● Rich Internet Application: This reference architecture is oriented toward the development of applications with a rich user interface that runs inside a web browser. This option was discarded because there was no plugin to run the application. The plugins are used to access the options in the application itself.
Logically structure the server part of the system using Service Application reference architecture.	Service application expose services that are consumed by other application. The Service Application was the best reference architecture to meet the requirements.
Physically structure the application using the Three-tier deployment pattern	Since the system is accessed from a web browser, we need to have a database server. Therefore the three-tier deployment is the most appropriate
Build the user interface of the client application using HTML/CSS and JavaScript	Best to make a clean simple UI, that everyone can use, no matter their experience in technology. Therefore these languages are the best fit and meet the requirements

Deploy the application using AWS	AWS will be used to deploy the CMS system.

Step 5: Instantiate architectural elements, allocate responsibilities, define interfaces

Design Decision and Location	**Rationale**
Create a module dedicated to access the database of the Service Application reference architecture	The service agents component from the reference architecture is adapted to access the database. This will ensure the achievement of QA-1, QA-5 and QA-6.

Step 6: Sketch views, record design decisions



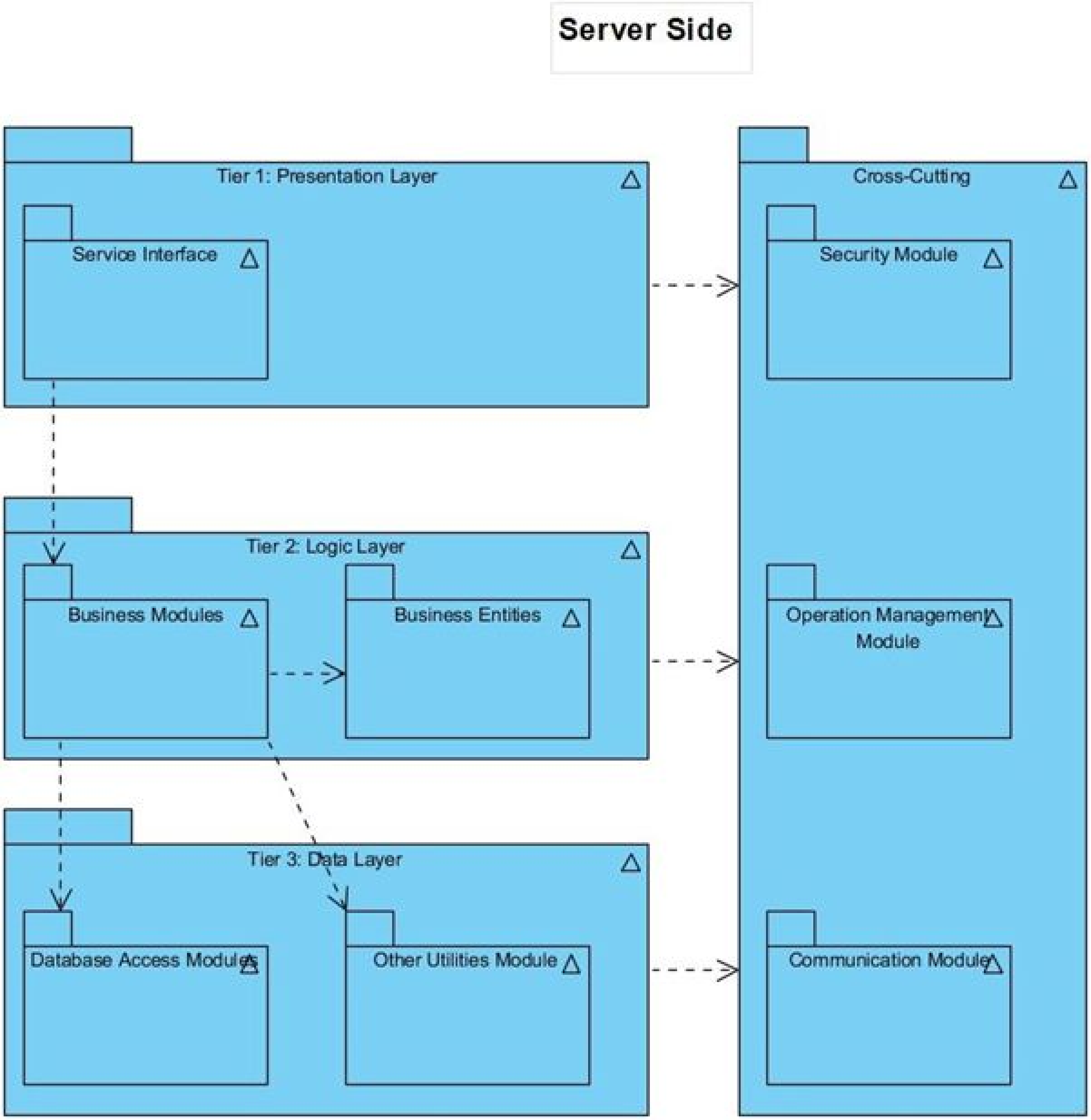


Figure 2: Client and Server side view of the system

Element	Responsibility
Presentation Layer (Client Side)	This layer contains modules that control UI and deals with input and output for

	users and communicates with the Business Layer
Business Layer(Client Side)	This layer contains modules that perform business logic operations that can be executed locally on client side. This layer also communicates with Data layer.
Data Layer (Client Side)	This layer contains modules responsible for communication with server, and store in database
UI Modules	Render for UI and receive inputs
UI Process Modules	Control flow of all system use case
Business Modules	Implement the business operations
Business Entities	Make up the domain model.
Server Request	Communicate with database to request information
Presentation Layer (Server Side)	This layer contains modules that expose services that are consumed by clients
Logic Layer (Server Side)	This layer contains modules that perform business logic operation that require processing from server side.
Data Layer (Server Side)	This layer contains modules that are responsible for data persistence and for communication

Service Interface	Modules expose service that are consumed by clients
Business Modules	Implement business operations
Business Entities	Make up the domain model
Database Access Module	Responsible for persistence of business entities into the relational database. It performs object oriented to relational mapping and shields the rest of the application from persistent details.
Other Utilities Module	Implement other utilities of the data layer

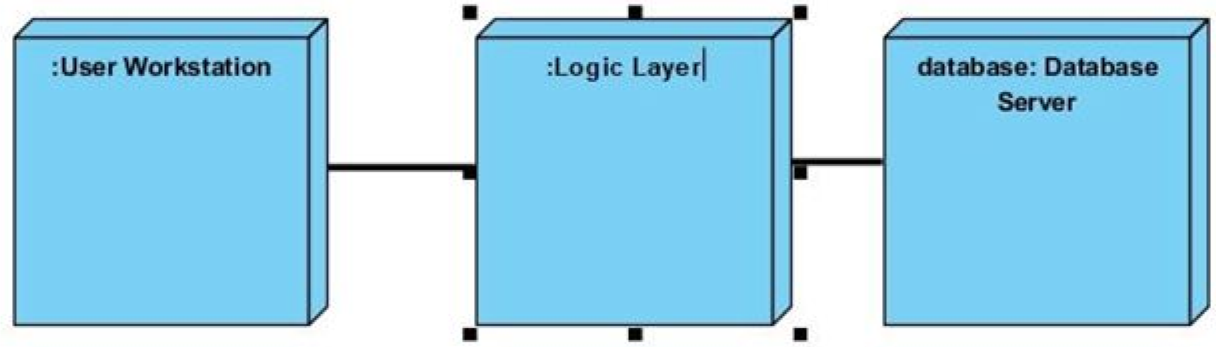


Figure 3: Initial deployment diagram of system

Element	**Responsibility**
User Workstation	User browser which host client side logic of system
Logic Layer	Communicating with user workstation and Database server

Database server	Host relational database

Relationship	Description
Between User Workstation and Logic layer	Any business logic that user has in interface is calculated here
Between Logic Layer and Database Server	Logic layer communicates with Database Server using SQl

Step 7: Perform analysis of current design and review iteration goal and achievement of design process

The following table summarizes the design progress using the Kanban board technique

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions made during the Iteration
	UC-1		Selected Reference Architecture establish modules that will support this functionality.
	UC-3		Selected Reference Architecture establish modules that will support this functionality

	UC-4		Selected Reference Architecture establish modules that will support this functionality
	QA-3		Identification of elements derived from deployment pattern that will need to be replicated
QA-4			No relevant decisions made
	CON-1		3 Tier architecture was used to support multiple users to connect to business layer. Decisions for concurrent access has not been made yet.
		CON-2	Database layer from three tier architecture was used so relation database could be created. Can access the database from data layer of architecture.

CON-3			No relevant decisions made
CON-4			No relevant decisions made
CON-5			No relevant decisions made
CON-6			No relevant decisions made

Iteration 2

ADD Step 1: Review Inputs

<table>
<tr><th>Category</th><th>Details</th></tr>
<tr><td>Design Process</td><td>This is a greenfield system from a novel domain. The purpose is to produce a sufficiently detailed design to support the construction of the system.</td></tr>
<tr><td>Primary Functional Requirements</td><td>From the uses cases presented from the first iteration, the primary uses cases are
●</td></tr>
<tr><td>Quality Attribute Scenarios</td><td><table><tr><td>Scenario ID</td><td>Importance to Stakeholders</td><td>Difficulty of implementation</td></tr><tr><td>QA-1</td><td>High</td><td>High</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>QA-2</td><td>Medium</td><td>Medium</td></tr>
<tr><td>QA-3</td><td>High</td><td>High</td></tr>
<tr><td>QA-4</td><td>High</td><td>Medium</td></tr>
<tr><td>QA-5</td><td>High</td><td>High</td></tr>
</table>
Only QA-1, QA-3, QA-4, QA-4, QA-5 were selected as drivers</td></tr>
<tr><td>Constraints</td><td>All constraints discussed in first deliverable are included as drivers</td></tr>
<tr><td>Architectural Concerns</td><td>All the architectural concerns discussed in first deliverable are included as drivers</td></tr>
</table>

Step 2: Establish iteration goal by selecting drivers

The goal of iteration 2 is to address the general architectural concern of identifying structures to support primary functionality. The developers consider the system primary uses cases:

- UC-1: Interact with the system
- UC-2 & UC-5: Push both public and private content onto projected services
- UC-3: Project profiles onto projectors/tvs using 3rd party hardware (ie. Chromecast)

Step 3: Choose one or more elements of the system to refine

The elements that will be refined in this iteration are modules located in different layers of the three tier web application, especially in the database and business logic layer. In general, the support of functionality in this system requires the collaboration of components associated with modules that are located in the different layers.

Step 4: Choose one or more design concepts that satisfy the selected drivers

In this iteration, several design concepts—in this case, architectural design patterns—are selected from the book Pattern Oriented Software Architecture, Volume 4. The following table summarizes the design decisions.

Design Decisions and Location	**Rationale and Assumption**
Create a **Domain Model** for the application	Before starting a functional decomposition, it is imperative to create an initial domain model for the system. This identifies major entities in domain, along with their relationships. There are no good alternatives. A domain model must eventually be created, or it will appear in sub/non-optimal fashion, leading to an ad hoc architecture that is hard to understand and maintain.
Identify **Domain Objects** that map to functional requirements	Each distinct functional element of application needs to be encapsulated in self-contained building block which is a domain object. Another possibility is to not consider domain objects but directly decompose the layers into modules. However this will increase the risk of not considering a requirement.
Decompose **Domain Objects** into general and specialized **Components**	Domain objects represent complete sets of functionality. This functionality however is supported by finer-grain elements located in within the layers. The “components” in this pattern are called modules.

Step 5: Instantiate architectural elements, allocate responsibilities, and define interface

Design Decisions and Location	Rationale
Create only and initial domain model	The entities that participate in the primary uses cases need to be identified and modeled but only an initial domain model is created. This is to accelerate this phase of the design.
Map the system uses cases to domain objects	An initial identification of domain objects can be made by analyzing the system's use cases. To address CRN-3, we identify domain objects for all use cases presented in iteration 1
Decompose the domain objects across the layers to identify layer-specific modules with explicit interface	This technique ensures that modules that support all the functionalities are identified. The architect will perform this task just for the primary use cases. This will allow another team member to identify the rest of the modules, thereby allocation work among team members. After establishing a set of modules, the architect realizes the need to test these modules. So a new architectural concern is identified. ● CRN-4: Need to test a majority of the modules Only a majority of the modules are covered by CRN-4 because the modules that implement the UI is difficult to test independently.

Step 6: Sketch views and record design decisions

As a result of the decisions made in step 5, several diagrams are created.

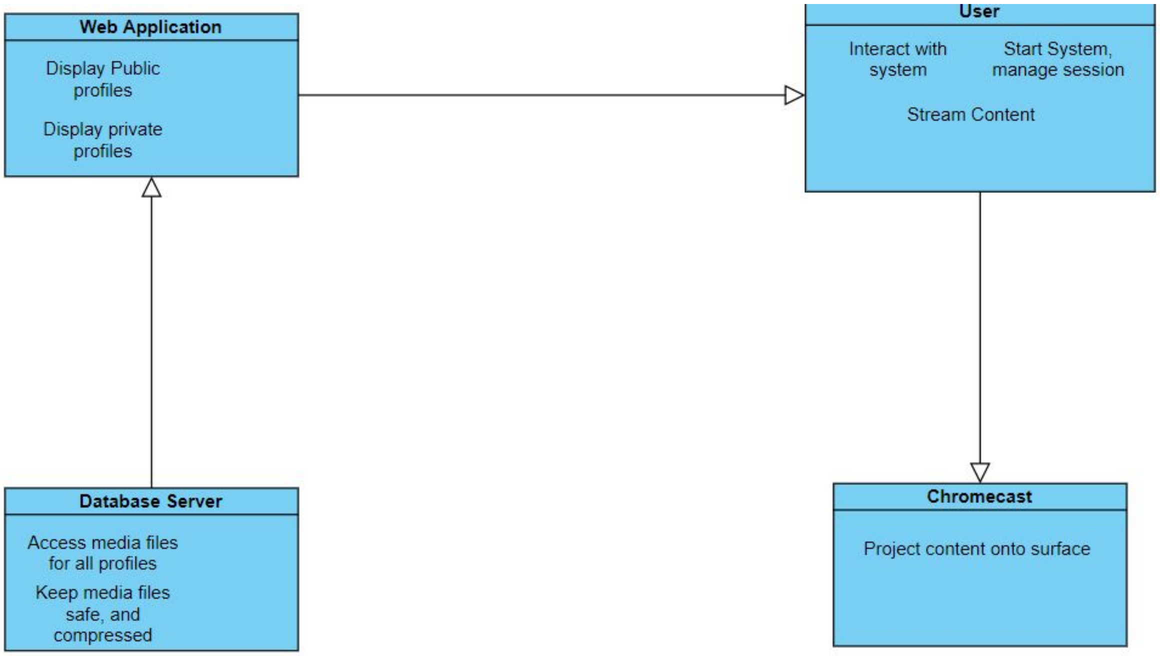

Web Application
Display Public profiles
Display private profiles
User
Interact with system
Start System, manage session
Stream Content
Database Server
Access media files for all profiles
Keep media files safe, and compressed
Chromecast
Project content onto surface

Figure - Initial Domain Model

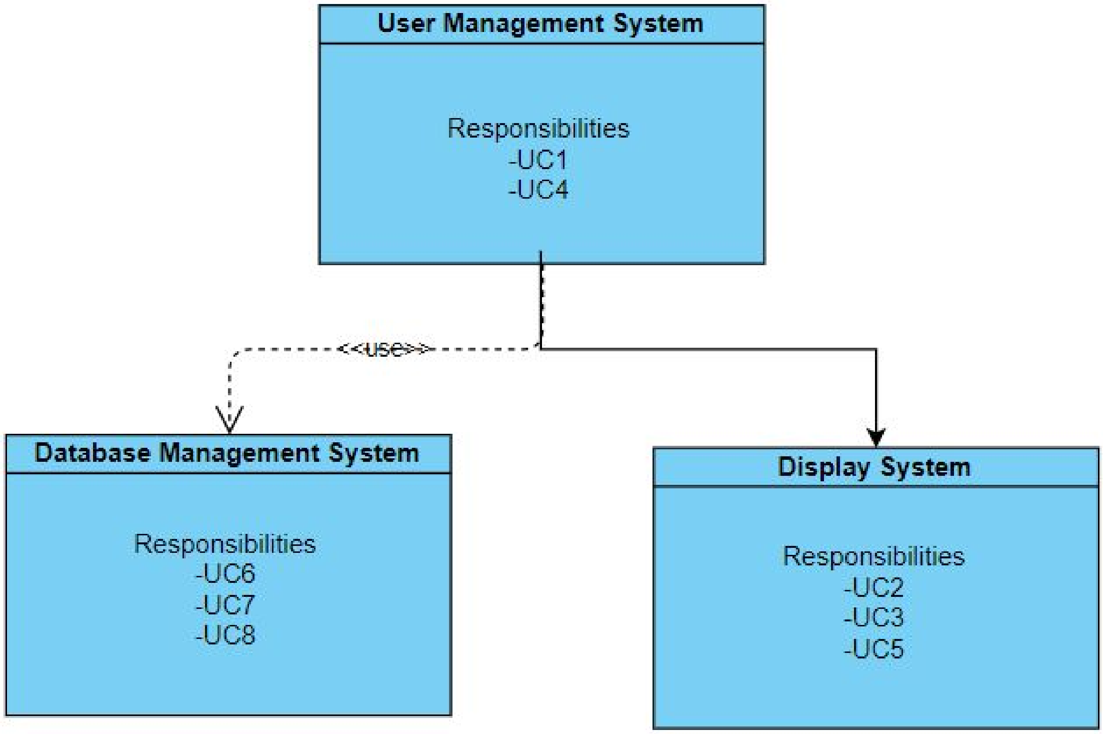


Figure- Domain objects associated with the use cases

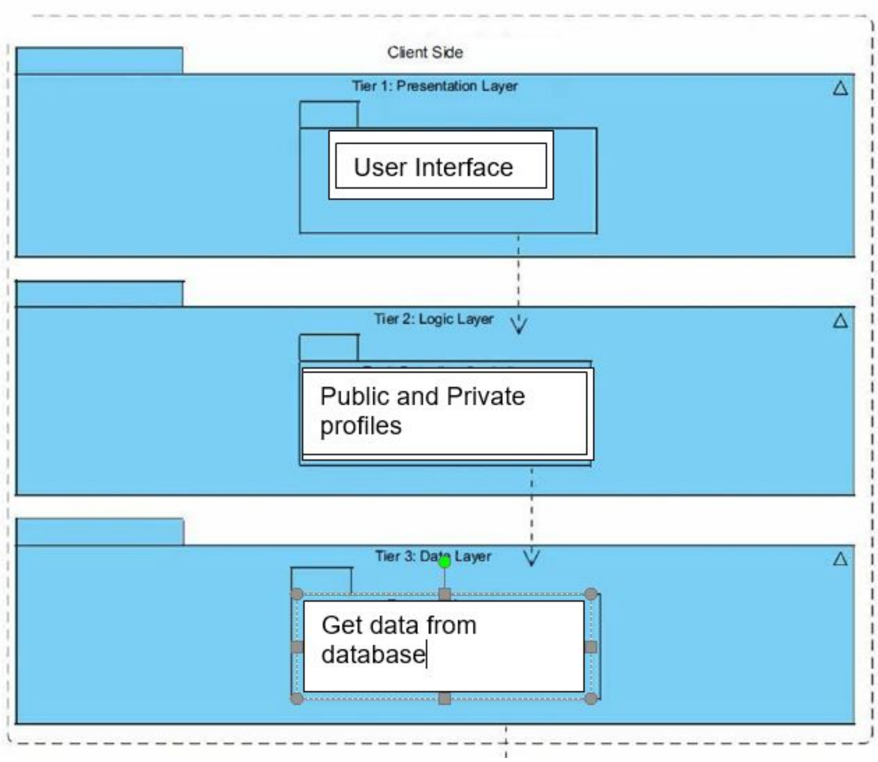
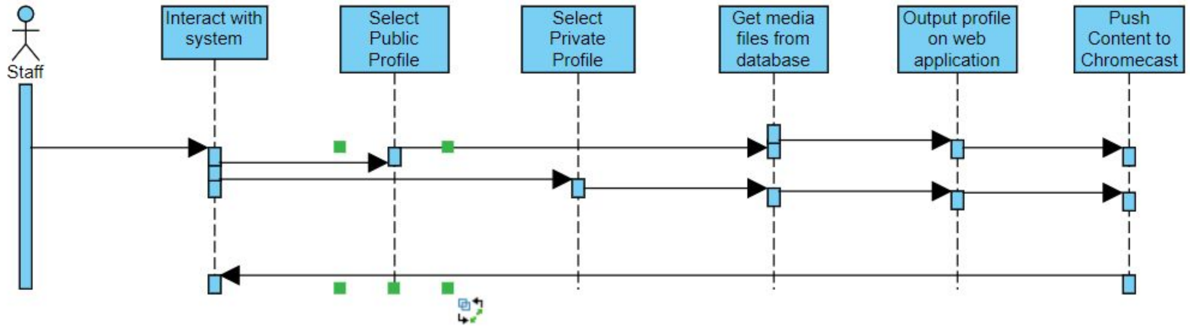


Figure - Modules that support the primary use cases

Elements	**Responsibility**
User Interface	This will be the place where users (staff), can interact with the system
Logic Layer (Public and Private profiles)	This will show the public and private profiles. Media files can be searched by tags. Also image processing algorithms can be used to sort images.
Data Layer	This will be the database that will store the media files.



Staff
Interact with system
Select Public Profile
Select Private Profile
Get media files from database
Output profile on web application
Push Content to Chromecast

Figure - Sequence Diagram for Use Cases 1, 2, 3, 4 & 5

The figure above shows the sequence diagram for uses cases 1, 2, 3, 4 and 5. When the staff first logs in, they will interact with the system. User will have the option to select the public or private profiles. Whatever actions will dictate what media files to query from the database. The media files will be then shown on the web application. After this the content can be pushed from the web application, to a projector or TV using the ChromeCast. The user can then interact with the system, by starting/stopping slideshow, or start audio stream etc.

Name	**Description**
System Interaction	User (staff) will need to login to the system. They can also start or stop the system, push the content to the projector, start slideshow etc.
Public Profile	Users can pick a public profile which will be used in a group setting session.
Private Profile	Staff can select a personal patient profile, which will be used for an individual therapy session.
Database Query	When a specific profile is chosen, the database will be queried, and will search for the media files associated with that profile.
Web Application Display	After the media files have been chosen, they will be displayed to the web page.
Chromecast to wirelessly display	Users can stream the web page wirelessly to a projector or TV.

Step 7: Perform analysis of current design and review iteration goal and achievement of design process

Established	Not Established	Partly Established	Design Decisions made during Iteration
UC- 1			Interfaces and modules have been identified in relation to this use case
UC-2			
UC - 3			Interfaces and modules have been identified in relation to this use case
UC - 4			Interfaces and modules have been identified in relation to this use case
UC-5			
		QA-1	Many users can use system concurrently

		QA-3	If a failure of malfunction does occur, the system has been designed to shutdown and restart.
		QA- 4	Data is queried from database.
	QA- 5		Design decisions have not been made in this iteration
	CON -1		Design decisions have not been made in this iteration
CON -2			Servers have been established in order to use system databases for the system
		CON -3	Design decisions have been made, ,but need expansion
CON -4			Modules for users to access data for all profiles have been established.

		CON - 5	Design decisions have not been made for this iteration.
		CON-6	Design decisions have not been made in this iteration
	CRN -1	CRN -1	Architecture has been established and needs more implementation
CRN -2			Programming language is chosen by the average familiarity with the language in accordance of the team
CRN -3			Workload has been divided.

Iteration 3

ADD Step 1: Review Inputs

Category	**Details**
Design Process	This is a greenfield system from a novel domain. The purpose is to produce a sufficiently detailed design to support the construction of the system.
Primary Functional Requirements	From the uses cases presented from the first

<table>
<tr><td></td><td>iteration, the primary uses cases are
●</td></tr>
<tr><td>Quality Attribute Scenarios</td><td><table>
<tr><th>Scenario ID</th><th>Importance to Stakeholders</th><th>Difficulty of implementation</th></tr>
<tr><td>QA-1</td><td>High</td><td>High</td></tr>
<tr><td>QA-2</td><td>Medium</td><td>Medium</td></tr>
<tr><td>QA-3</td><td>High</td><td>High</td></tr>
<tr><td>QA-4</td><td>High</td><td>Medium</td></tr>
<tr><td>QA-5</td><td>High</td><td>High</td></tr>
</table>
Only QA-1, QA-3, QA-4, QA-4, QA-5 were selected as drivers</td></tr>
<tr><td>Constraints</td><td>All constraints discussed in first deliverable are included as drivers</td></tr>
<tr><td>Architectural Concerns</td><td>All the architectural concerns discussed in first deliverable are included as drivers</td></tr>
</table>

Iteration 3: Addressing Quality Attribute Scenario Driver (QA-3)

This section presents the results of the activities that are performed in each of the steps of ADD in the third iteration of the design process. Building on the fundamental structural decisions made in iterations 1 and 2, we can now start to reason about the fulfillment of some of the more important quality attributes. This iteration focuses on just one of these quality attribute scenarios

Step 2: Establish iteration goal by selecting drivers

For this iteration 3, the developers focused on QA-3 quality attribute scenario. A failure occurs in the system during an operation. The system will need to shut down and restart immediately.

Step 3: Choose one or more elements of the system to refine

For this particular scenario, the elements that need to be refined are:

- Application Server
- Database Server

Step 4: Choose one or more design concepts that satisfy the selected drivers

Design Decisions and Location	**Rationale and Assumption**
Introduce active redundancy, by replication application server and other critical components such as database.	When replicating these critical elements, the system can withstand the failure of one of the replicated elements without affecting functionality.
Backup database regularly	In case the database server crashes, due to malfunction or such. Having backuped data will ensure previous data, and prevent data loss.
Introduce element from **message queue** technology family	Traps received from time servers are placed in the message queue and then retrieved by the application. Use of a queue will guarantee that traps are processes and delivered in order (QA-1)

Step 5: Instantiate architectural elements, allocate responsibilities, and define interfaces

The instatintion design decisions are summarized in the following table:

Design Decisions and Location	**Rationale**
Deploy Message queue on separate node	Deploying the message queue on separate node will guarantee that no traps are lost in case of application failure. This node is replicated using the tactic of active redundancy, but only one copy receives and treats events coming from the CMS system.
Use active redundancy and load balancing in the application server	Load balancer needs to be deployed to regulate traffic for multiple application server. This will introduce a new architectural concern CRN-4: Manage states in replicas.
Implementing load balancing and redundancy using support	There are many technological options for load balancing and redundancy that can be implemented without having to develop ad hoc solution that is harder to support and less mature

Step 6: Sketch Views and Record Design Decisions

Figure 1 shows the refined deployment diagram that includes the introduction of redundancy in the system.

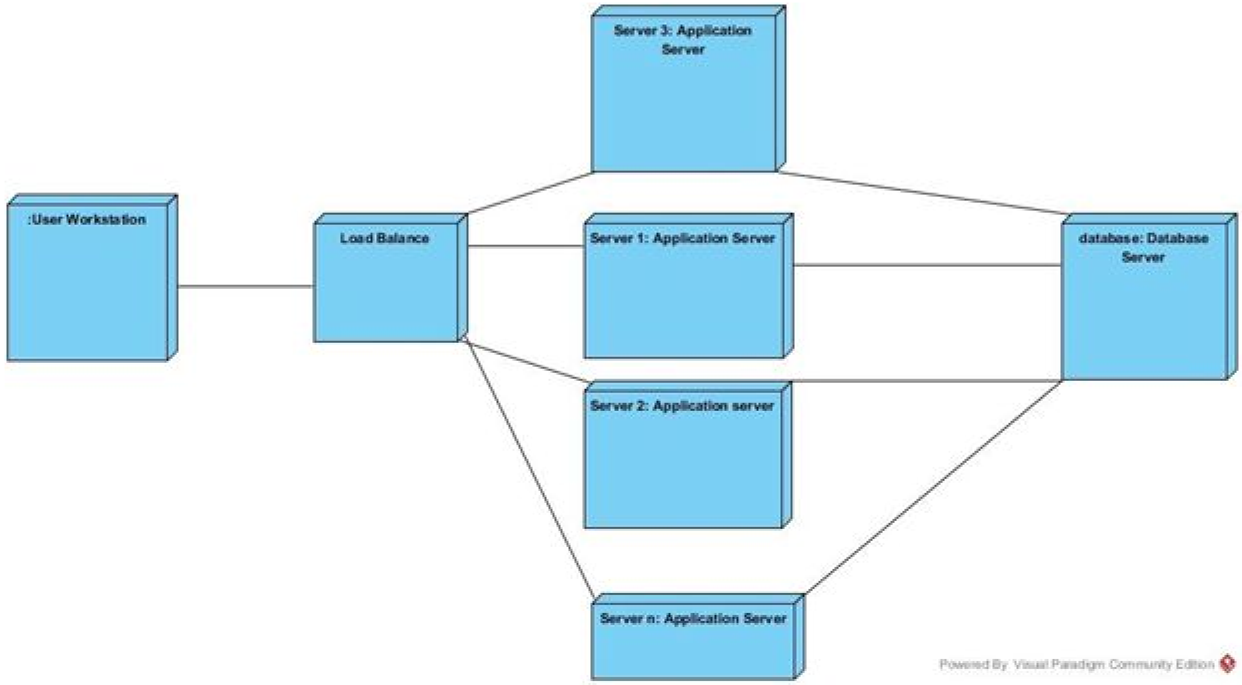

Server 3: Application Server
:User Workstation
Load Balance
Server 1: Application Server
database: Database Server
Server 2: Application server
Server n: Application Server
Powered By Visual Paradigm Community Edition

Figure 1: Deployment Diagram

Element	**Responsibility**
Load Balance	Takes the requests coming from users to application servers.
Database Server	Takes control of all the databases needed for the system.

Step 7: Perform analysis of current design and review iteration goal and achievement of design purpose

Established	Not Established	Partly Established	Design decisions made during iteration
QA-1			Performance of system should be scanned periodically to prevent any malfunctions. Also

			data should be logged at certain intervals to prevent data loss, if performance is suffering due to data loss.
QA-5			System for tracking by user ID has been established
CON-1			Multiple servers are established a co-related through a database to improve flow of traffic and maintain load.
CON-3			Database has been established for data recovery
CON-6			Restriction and security protocols are established for secondary systems

2. Conceptual System Design

The reminiscence therapy system consists of a database, a web portal, and a media playing device. The web portal accesses the data stored in the database based on who is logged in, the web portal also transmits data to a media playing device, in this case a ChromeCast. The basic layout of the system is displayed in the diagram below.

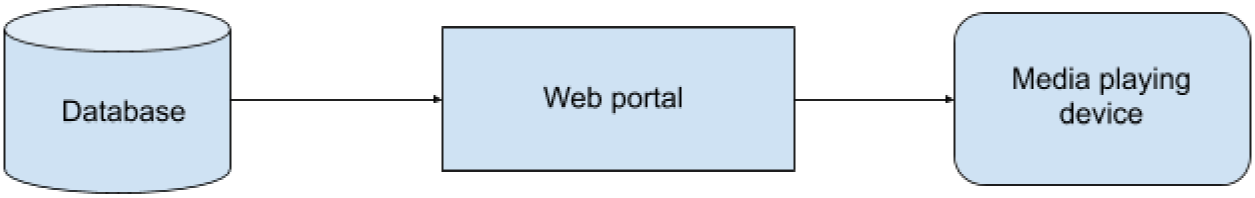


Figure 2.1 Basic Layout of System

The database will stream media including photos, videos, and audio files through the web portal to the ChromeCast whenever a user requests a stream. The transfer rate will be limited by the network the system runs on. The web portal allows users to view the media stored on the database, the users can see the names of files as well as thumbnails which preview the file's contents. The names and thumbnails are transferred to the web portal when a user logs in.

The database is used to store reminiscence media uploaded by caregivers and Ontario Shores employees. The web portal is used to access the media from the database, it is also used to send data to a ChromeCast. The ChromeCast is connected to a display (projector, T.V.) and used to display the media from the database to said display.

3. Definition of Integration Tests

The system consists of multiple parts which must work together in order to perform a specified task. At a higher level the system is split into large components dealing with data access and data display, since these 2 parts are modular they can be tested separately. We will employ the bottom-up approach to testing the system, ensuring the smaller parts are working correctly before they are merged as part of a larger system.

The data access portion deals with establishing a database connection, interfacing with the user, downloading the requested data, and uploading user data. The first test would be to establish a database connection, then we will need to test the upload and download functions. Once the database connection and methods are functioning as intended, the next step is to test the UI that users will interact with. The various functions provided by the UI including viewing the files belonging to the profile, uploading files, and displaying files will be tested.

The second part to be tested is the media display portion of the system. We would need to test the transmission of data wirelessly from a computer to a display, and also ensure that the requested data is being downloaded correctly.

Once both parts have been tested,we will test them together and ensure they interface correctly with each other.

4. Estimated Cost

Table 4.1 Time and Effort Estimation

Resource Skill Set	Development Timeline (in business days)
UI/UX	**2**
Web Developer 1	**10**
Web Developer 2	**10**
QA Testing	**3**
Database Engineer	**5**
Total Project Timeline/Cost	**30**

Table 4.2 Cost Estimation

Item	Quantity	Cost Per Unit (Pre Tax)
Optoma W412 Projector	2	$819.00
Google Chromecast	3	$45.00
Total		**$1,773.00**

COCOMO

This project will be an organic project, because our team size is small (4 people), the problem is well understood, and the team members have nominal experience in solving this type of problem. We estimate to have at most 1000 lines of code. We can estimate the basic model with the formula $E= a(KLOC)^b$. Using this model, and knowing the values for a and b being 2.4 and 1.05 respectiley, our estimate of person months is E = 2.4(1)^1.05, our estimate E = 2.4 person-months.

5. Updated Project Plan

The project will be split between two semesters. Requirements gathering and initial prototype will be done during the first semester. Meanwhile the final product deployment will be done in the second semester. Detailed plan included below.

Table 5.1 Project schedule

Task	Start date	(Tentative) Finish date	Actual Finish Date
Progress Report 1	2019-09-15	2019-10-24	2019-10-24
Project topic	2019-09-15	2019-09-20	2019-09-20
Background research	2019-09-21	2019-09-27	2019-09-27
Stakeholders requirement	2019-15-09	2019-10-11	2019-10-11
Planning	2019-10-06	2019-10-19	2019-10-19
Finalize report	2019-10-17	2019-10-24	2019-10-24
Progress Report 2	2019-10-25	2019-11-14	2019-11-14
Finalize budget	2019-11-07	2019-11-08	2019-11-14
Developing Prototype	2019-10-12	2019-11-14	2019-11-23
Design	2020-01-10	2020-02-01	2020-02-01
Develop	2020-01-15	2020-02-15	2020-02-01
Testing	2020-02-01	2020-02-19	2019-11-24
Touch ups and finalization	2020-02-19	2020-03-21	2019-11-27 - Presentation 1 TBA- Semester 2
Capstone	2019-11-28 - S1	2019-11-28 - S1	2019-11-28 -

Presentation	TBD - S2	2020-04-10 - S2	Semester 1 TBD Semester 2

6. Contribution Matrix

The list of tasks and contributors are shown in Table 6.1.

Table 6.1: Contribution Matrix

	People			
Members	Abdurrahman Ansari	Mohammed Hameeduddin	Dhanushga Lionel	Mingwei Zhang
Solution Research	25%	25%	25%	25%
System Design and Flow	0%	50%	25%	25%
UI Modelling	25%	0%	25%	50%
Report 2	30%	20%	30%	20%