Logo

Description automatically generated

University of Bahrain

College of Information Technology

Department of Computer Science

ITSE 305: Software engineering project  
BSc in Software Engineering

**Phase 3 – Balagh**

**System**

**Group Members:**

1. Maha Mohammed Ali: 202002565
2. Musherah Moqbel Ali :202002276
3. Fatema Salman Abbas :202005779
4. Noor Jaafar Ali: 202007922
5. Abdullah Aktham Khaleel: 202004678

Table of Contents

[1. Executing 2](#_Toc133689037)

[1.1 System Requirements 3](#_Toc133689038)

[1.1.1 Design Purpose 3](#_Toc133689039)

[1.1.2 UML Model 4](#_Toc133689040)

[1.1.3 Define Quality Attributes Scenarios 5](#_Toc133689041)

[1.1.4 Quality Attribute List 6](#_Toc133689042)

[1.1.5 Catalog Constraints 8](#_Toc133689043)

[1.1.6 Priority List 9](#_Toc133689044)

[1.1.7 Concerns 9](#_Toc133689045)

[1.2 Design Process 9](#_Toc133689046)

[1.2.1 Attribute Driven Design 10](#_Toc133689047)

[1.2.2 Iteration 1: Establishing an Overall System Structure 11](#_Toc133689048)

[1.2.3 Iteration 2: Identifying Structures to Support Primary Functionality. 19](#_Toc133689050)

[1.2.4 Iteration 3: Addressing Quality Attribute Scenario Drive (qa-6) 27](#_Toc133689051)

[2. Milestone Reports 30](#_Toc133689052)

[3. Performance Reports 32](#_Toc133689053)

# 1. Executing

The phase of execution is a cornerstone of the project management process where the project team must take essential measures to actualize the tasks defined in the project plan. This stage involves the creation of the tangible outputs, intangible services, or consequential results of the project, and usually necessitates the allocation of the most resources to achieve. The execution phase necessitates coordinating and guiding members of the project team, managing resources, and communicating with stakeholders to ensure that the project is advancing in alignment with the plan. To maintain the project's alignment and address any issues or modifications promptly, project managers must closely observe the execution phase.

|  |  |  |
| --- | --- | --- |
| Knowledge Area | Executing Process | Outputs |
| *Project Integration Management* | Guide and manage project work | Deliverables  Work Performance Data  Change Requests  Project Management Plan Project document Updates |
| *Project Quality Management* | Perform Quality Assurance, Control Quality | Quality Management Plan  Quality Metrics  Quality Checklists  Work Performance Data  Change Requests |
| *Project Human Resource Management* | Acquire Resources, Develop Team, Manage Team | Project Staff Assignments  Resource Calendars  Team Performance Assessments  Change Requests  Project Management Plan Updates |
| *Project Communications Management* | Manage Communications | Project communications  Project documents updates Project management plan updates  Organizational process assets updates |
| *Project Procurement Management* | Conduct Procurements, Control Procurements | Procurement Management Plan  Resource calendars Change requests Project management plan updates  Project Management Plan Updates |
| *Project Stakeholder Management* | Manage Stakeholder Engagement, Identify Stakeholders | Stakeholder Register  Stakeholder Management Plan  Issue Logs  Change Requests  Project Management Plan Updates |

Table 3.1S Executing process and outputs.

## 1.1 System Requirements

### 1.1.1 Design Purpose

As technology continues to permeate every aspect of our daily routines, it serves as an ever-increasing conduit between users and systems. Balagh System stands to revolutionize our lives, providing stiff competition to other systems. By enabling citizens and residents to report issues or errors, seek information, offer suggestions, and interact with decision-makers across various ministries, Balagh application offers significant potential to enhance our quality of life. User submissions are sorted based on the number of ministry employees, with designated staff handling each request, ensuring that the app offers custom made solutions to users.

The Balagh System is comprised of four primary components: Account, Services, Contact, and Complaints Management. Users can create accounts and register using CPRs, while basic users must provide their email addresses and phone numbers for contact purposes. The app interface provides dedicated windows for employees and users, offering customized services and functionalities for each user category.

The Services component encompasses the central functionalities the system provides to citizens and residents, empowering them to submit suggestions, seek information, and report complaints. Contact feature bridges communication between users and ministry employees, facilitating swift resolution of issues, provision of solutions, and responses to inquiries and suggestions.

Complaints Management feature is designed specifically for employees, enabling them to oversee and categorize a multitude of submissions based on demand status and type. In addition, employees can leverage the Report Management content management system to manage user complaints, inquiries, and suggestions efficiently.

### 1.1.2 UML Model

Diagram

Description automatically generated

### 1.1.3 Define Quality Attributes Scenarios

|  |  |
| --- | --- |
| **UC-1 Login** | To ensure the security of the system, citizens and residents must provide their CPR and mobile number upon logging in. A one-time password (OTP) is then sent to their registered mobile number to authenticate their access. Once successfully authenticated, users can proceed to report issues, seek information, or avail any services offered by the app. On the other hand, Employees are required to enter their CPR and ID number for system access. |
| **UC-2 Register** | To access the system, end-users such as citizens, residents, and ministry employees must create an account by registering. The registration process requires employees and residents to enter their CPR number, email address, and mobile phone number. However, the specific registration requirements for ministry employees may vary. In such cases, they are required to enter their ministry name, CPR number, and mobile number to activate their account.  Once the account is activated, users can log in using their CPR number and password for citizens or their ID number and mobile number for employees. It is recommended that users remember their login credentials for future use. |
| **UC-3 Suggestions** | Users can submit their suggestions and ideas to any ministry of their choice by writing a proposal and specifying the relevant ministry. Once the proposal is complete, it can be submitted to the concerned ministry for consideration. |
| **UC-4 Inquiry** | Both citizens and residents can send inquiries through the system regarding unclear laws or regulations, violations, awareness campaigns, or any other related matter. To do so, users can click the "+" button, select the "Inquiry" option, and then proceed to write and submit their inquiry through the system. This feature enables users to seek relevant information and clarification from the concerned ministries. |
| **UC-5 Complaint** | Users can submit their complaints or reports to any ministry of their choice through the system. To do so, users must specify the relevant ministry, identify the location of the violation, and provide a description of the complaint. In addition, users have the option to attach pictures, videos, or documents to support their report. This feature enables users to report any violations or issues they encounter, ensuring that the relevant authorities can take appropriate action. |
| **UC-6 Live Chat Contact** | Users can engage in a live chat with the specialized staff appointed by the relevant ministry to discuss their complaint or suggestion and track its progress or status. To do so, users can select the relevant ministry and then choose the live chat option. This feature enables users to communicate directly with the designated employee and receive updates on the status of their complaint or suggestion. |
| **UC-7 Customer Services Contact** | If a user encounters any problems while using our system, they can easily contact our support team for assistance. Users can select the "Contact Team" option from the side menu, and a chat session will open where they can message us. Additionally, the system provides an audio call option to make it even easier for users to communicate with our support team and resolve any issues they may be experiencing. |
| **UC-8 Complaints management** | The system will have a dedicated window for employees of different ministries to manage various inquiries, complaints, and suggestions. This window will only be accessible to employees who are registered in the system. They can accept or reject complaints and inquiries, as well as communicate with normal users to provide updates or information. |

### 1.1.4 Quality Attribute List

|  |  |  |  |
| --- | --- | --- | --- |
| **QA-1** | Useability | Users can use the system with ease after only 15 minutes of training, and they can create complete complaints, suggestions, or inquiries without any help from trainers. Similarly, employees can understand the functionality of the system and how the CMS works in less than three hours. | **All** |
| **QA-2** | Performance | The system is capable of transferring and collecting user information within 3 seconds and presenting it to the relevant employee in the corresponding ministry. | **UC-8** |
| **QA-3** | Security | The system will utilize strong cryptography through the RSA algorithm to ensure that unauthorized individuals are unable to access the information of registered users. | **UC-1** |
| **QA-4** | Modifiability | The system has been designed with modularity in mind, which allows developers to easily add new functionalities in the future. Each portion of the system is independent, so any changes or modifications can be made quickly and cost-effectively without affecting the overall functionality of the system. | **All** |
| **QA-5** | Portability | The system will have a mobile application architectures designs, and will be run in Android OS and IOS to serve as many users as possible and the source code of the system was designed to move from one platform to another with small changes. | **All** |
| **QA-6** | Availability | The system will be able to work 24/7 because it has different backup servers, so when one has a problem the second one will work, and the first one will try to fix the problem and it will restart without losing any data. | **All** |
| **QA-7** | Scalability | The system was able to handle more than 4000 concurrent users, and the type of infrastructure accepts any development in hardware and software for next improvement. | **Uc-7** |
| **QA-8** | Safety | The system will display an error message if the user attempts to submit a duplicate report with identical information within the same week. This is to allow the ministries to address the issue, and if no response is received, the user will be able to submit the report in the following week. | **UC-3,**  **UC-4,**  **UC-5** |

### 1.1.5 Catalog Constraints

|  |  |
| --- | --- |
| **ID** | **Constraint** |
| **CON-1** | Flutter will be used to develop the mobile application, which will support both iOS and Android operating systems. |
| **CON-2** | The system is designed to handle a large number of users, approximately 30,000 users simultaneously, and ensure they receive a good user experience. This will lead to improved system performance. |
| **CON-3** | The system will store all events from the last 30 days to ensure that any issues or errors can be handled effectively, even if a user's complaint or report has been lost. This data retention policy will ensure that important information is not lost and can be retrieved as needed. |
| **CON-4** | The system modifications will be carried out only during off-hours to avoid any disruption in user experience. Additionally, users will be informed beforehand of any maintenance work that needs to be done. |
| **CON-5** | The system will have multiple backup servers to ensure high availability and prevent any interruptions or data loss in case of errors or problems. This will help to maintain a seamless user experience and avoid any inconvenience for the users. |
| **CON-6** | The system should be designed to support cross-platform compatibility, meaning it should be accessible on various platforms including mobile, to provide the users with flexibility in accessing the system. |

### 1.1.6 Priority List

|  |  |  |
| --- | --- | --- |
| **Scenario ID** | **Importance to the customer** | **Difficulty of implementation according to the architect** |
| **QA-1** | high | medium |
| **QA-2** | high | high |
| **QA-3** | medium | low |
| **QA-4** | high | high |
| **QA-5** | medium | medium |
| **QA-6** | high | high |
| **QA-7** | high | high |
| **QA-8** | medium | low |

### 1.1.7 Concerns

|  |  |
| --- | --- |
| **ID** | **Concerns** |
| **CRN-1** | Interduce working hours for the development team within the first day of starting the project and allocate the work. |
| **CRN-2** | The development team assigned to the project must possess expertise in various programming languages and frameworks, such as, libraries, Java for Android, and C# for iOS platforms. This is necessary to meet the requirements of the project and develop the system effectively across multiple platforms. |
| **CRN-3** | We need to create a preliminary system structure, which will help us to quickly define the system requirements and provide stakeholders with a tangible system to test. |

## 1.2 Design Process

### 1.2.1 Attribute Driven Design

#### 1.2.1.1 Step1: Review Inputs

|  |  |
| --- | --- |
| **Category** | **Details** |
| Design purpose | This is a greenfield system from a mature domain. The purpose is to produce a sufficiently detailed design to support the construction of the system. |
| Primary functional requirements | From the use cases presented in Section 2.1, the primary ones were determined to be:  UC-3: Because it directly supports the core business.  UC-4: Because it directly supports the core business.  UC-5: Because it directly supports the core business.  UC-8: Because it directly supports the core business. |
| Quality attribute scenarios | The scenarios were described in Section 2.2 They have now been prioritized (as discussed in Section 2.2.1) as follows:   |  |  |  | | --- | --- | --- | | Scenario ID | Importance to the customer | Difficulty of implementation according to the architect | | QA-1 | high | medium | | QA-2 | high | high | | QA-3 | medium | low | | QA-4 | high | high | | QA-5 | medium | medium | | QA-6 | high | high | | QA-7 | high | high | | QA-8 | medium | low |   From this list, only QA-2, QA-4, QA-6, and QA-7 are selected as drivers. |
| Constraints | All the constraints discussed in Section 2.3 are included as drivers. |
| Architectural concerns | All the architectural concerns discussed in Section 2.4 are included as drivers. |

### 1.2.2 Iteration 1: Establishing an Overall System Structure

#### 1.2.2.1 Step 2: Establish Iteration Goal by Selecting Drivers

This is the initial design iteration of a new system, and the main goal is to address the architectural concern CRN-3, which is to establish the overall system structure as outlined in section 2.4. While the iteration is primarily focused on this architectural concern, the architect must also take into account all other factors that may impact the system's overall structure. This includes being mindful of the following:

QA-2: Performance

QA-4: Modifiability

QA-6: Availability

QA-7: Scalability

CON-1: Flutter will be used to develop the mobile application, which will support both iOS and Android operating systems.

CON-4: The system modifications will be carried out only during off-hours to avoid any disruption in user experience. Additionally, users will be informed beforehand of any maintenance work that needs to be done.

CON-5: The system will have multiple backup servers to ensure high availability and prevent any interruptions or data loss in case of errors or problems. This will help to maintain a seamless user experience and avoid any inconvenience for the users.

CRN-2: The development team assigned to the project must possess expertise in various programming languages and frameworks, such as, libraries, Java for Android, and C# for iOS platforms. This is necessary to meet the requirements of the project and develop the system effectively across multiple platforms.

Diagram

Description automatically generated

#### 1.2.2.2 Step3: Choose One or More Elements of the System to Refine

Once again, as this is a greenfield development effort, the element being refined is the entire Balagh system. Refinement in this case is achieved through decomposition.

#### 1.2.2.3 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 will be **Mobile application** reference architecture. | The mobile application reference architecture will help the client to access from different OS such as Android OS and IOS **(CON-1)** to have a rich client interface and it will help the poor devices to access to our system. Therefore, the application will be accessible everywhere through the internet. On the other hand, we can use mobile application architecture to handle devices that help different clients to access systems from different platforms.  This capability will help to archive **(QA-5).** |
|  | **Discarded alternatives:**   |  |  | | --- | --- | | **Alternative:** | **Reason for Discarding:** | | Rich Internet  applications  (RIA) | A rich user interface running inside a browser considers the role of this reference architecture, it very useful for loading time but the reason for discard them is that plug-in execution environments may not be available in all platforms which contradicts with our (CON-1). | | Rich Client Application (RCA) | Rich client application that downloads into user machine and this interface will provide high-performance, interactive rich user experience. Therefore, the reason for discarding this reference architecture is that the system works with no network connectivity, and this will affect user experience and availability **(QA-6)**, since client data is sent by internet connection in our system. | |
| Logically structure  the server part of  the system using the  **Service Application**  reference architecture | Reference architecture does not support interface and it is expose services for other applications, so it is very helpful to use it with our application.  No other alternatives were considered and discarded since this architecture is applicable to work with different reference architecture at the same time and it is considered by architect. |
| Physically structure  the application  using the **Load-Balanced Cluster** | Using the Load-Balanced Cluster pattern, the workload is distributed across multiple servers, client requests will be received by the load balancer which redirects them to the various servers according to their current load. Therefore, this deployment model will play an important role in **(QA-6)** and **(QA-2)** and distribute the load over different servers this will help achieve **(CON-2)(CON-4).** Moreover, reference helps in maintaining security since the data will be resident of the separate tires, so this will improve achieving of **(QA-3).** Other alternatives were discarded such as **two-tire** **three-tire** or **four-tire** deployment they support security, but they are not applicable with our solution because our system deal with a high amount of load and the same time our system must support availability, so we require deployment that oversee a high number of users. |
| Build the user interface of the client application using the **Dart Language** and other  **Dart libraries** and **flutter framework**. | Since dart language can be used to develop mobile applications, we can use it to develop our system to satisfy **(CON-1)** and **(CRN-2).**  Other alternatives were discarded because we would like to develop a system that is a cross platform which works on mobile applications **(QA-5)** and for this reason it was discarded**.** |
| Deploy the application  using the **mobile start technologies** | Access application will be obtained through mobile applications (Android OS and IOS) .Mobile will be able to run system with high portability **(QA-5)**, also since system is divided into portions this will support **(QA-4)** and **(CON-4).**  The alternative would be using of applets but the low capability of this type of architecture will negatively affect the system security, performance, and safety of our system. |

#### 1.2.2.4 Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces.

|  |  |
| --- | --- |
| **Design Decision and Location** | **Rationale** |
| Create a module dedicated to accessing the content management system **(CMS)** in the data layer of the service application reference architecture | This module will help ministry employees to manage access to complaints, inquiries, and suggestions and give them the needed tools and resources to oversee a minimum of 4000 simultaneous users **CON-2** and provide a simple interface that has necessary tools that support filtering complaints, inquiries, and suggestions. |
| Add an external infrastructure to the **mobile applications** reference architecture. | Some mobile devices do not have enough resources to manage a specific type of system processes, so adding external infrastructure (it may be an external server) will help to handle these processes then send the output to the mobile application to execute it directly. This will support **(QA-2)**, **(QA-6)** and **(CON-5)**. |

#### 1.2.2.5 Step 6: Sketch Views and Record Design Decisions.

Diagram

Description automatically generated

FIGURE 1: Reference Architecture

|  |  |
| --- | --- |
| **Element** | **Responsibility** |
| **Presentation CS** | This layer contains the modules that control the client interface and describe the interaction between customer and system. |
| **Business CS** | This layer preforms business logic operation that can be executed locally on the client side or externally throw the server side. |
| **Data CS** | This layer contains modules that responsible of communication between customer side and server side in mobile system . |
| **Cross-cutting CS** | This layer will be responsible about managing the modules and their functionalities that goes across different layers, such as security, logging, and I/O this part will help in achieving **QA-3.** |
| **UI modules** | This module renders the user interface and receives their inputs. |
| **UI process module** | This module responsible for controlling the flow of all system use cases (include the navigation between screens). |
| **Business logic CS** | This part responsible for implementing business operation that can preformed locally in mobile application or expose business functionality from the server side in mobile applications. |
| **Business entities CS** | These entities make up the domain model. They may be less detailed than those on the server side. |
| **Communication modules CS** | These modules consume the services provided by the application running on the server side in mobile application. |
| **Services side server SS** | This layer contains modules that expose the services that are consumed by the clients in mobile application. |
| **Business login SS** | This layer contains modules that preform business logic operation that require processing on the client side. |
| **Data SS** | This part will be responsible about the data persistence and for communication with mobile system. |
| **Cross-cutting SS** | This layer will manage the interaction between different layers especially the modules that have functionality, such as security, logging, and I/O. |
| **Service interface SS** | These modules expose services that are consumed by the clients in mobile application. |
| **Business modules SS** | This module responsible about implementing business operations that pass by the client side in mobile application. |
| **Business entities SS** | These entities make up the domain model. |
| **DB access module** | This module responsible for persistence of business entities into relation database. By preforming object-oriented and shields the rest of application from persistence details. |
| **Business workflow SS** | This component responsible about managing (long-running) business process, involve executing multiple use cases in mobile application. |
| **Application façade SS** | These modules will provide simplified interface to the business logic component in mobile application. |
| **Helpers and utilities SS** | This modules in the data layer can use these components, but they do not have any specific functionality in mobile application. |
| **Service agents SS** | This part abstract communication mechanisms used to transfer data to external services in mobile application. |
| **Data access SS** | This component responsible about data retrieval and storage handle, which provide persistence mechanisms and common operations in mobile application. |
| **Data sources** | This component will be responsible of storing and managing the data that transfers by mobile applications. |

**Table 2**

The responsibilities of the elements are summarized here:

|  |  |
| --- | --- |
| **Element** | **Responsibilities** |
| **User workstation )Mobile(** | Users' devices will be able to access the application server to perform business logic. |
| **Application server** | The server will host the server-side logic for the mobile application, which will assist in performing the necessary business logic. |
| **Fire wall** | The component responsible for monitoring incoming and outgoing network traffic will permit or block data packets based on a set of security rules. Additionally, it will perform network-level encryption using RSA to ensure data security **(QA-3)**. |
| **Load Balancer** | The system should be able to manage the load between devices to ensure that if one server fails to handle the load of many users, the load balancer can transfer some users to another server. This capability is necessary to address the system's continuity concern **(CON-2).** |

Also, information about the relationships between some elements in the diagram summarized in the following table.

|  |  |
| --- | --- |
| **Relationship** | **Description** |
| **Between application server and data sources** | Communication through HTTP protocol. |

Diagram

Description automatically generated

FIGUER 2: initial deployment of Balagh system

#### 1.2.2.6 Step 7: Perform Analysis of Current Design and Review Iteration.

|  |  |  |  |
| --- | --- | --- | --- |
| **Not Addressed** | **Partially Addressed** | **Completely Addressed** | **Design Decisions Made During the Iteration** |
|  | UC-1 |  | Selected reference architecture modules will support this functionality. |
|  | UC-2 |  | Selected reference architecture modules will support this functionality. |
|  | UC-6 |  | Selected reference architecture modules will support this functionality. |
|  | UC-7 |  | Selected reference architecture modules will support this functionality. |
| QA-1 |  |  | No relevant decision was made, that is necessary to identify the elements that participate in the use because that associated with the scenario. |
|  | QA-2 |  | Introduction to the content management performance model by identifying the load-balance cluster this will help achieving high performance of UC-8. Interface and details about the component not been defined yet. |
|  |  | QA-3 | Using a firewall in deployment model will help to encryption data that moves from one element to another. |
|  |  | QA-4 | By using two separated reference architectures this will reduce coupling in the system. So, the modifiability will achieve since there are no integrated components. |
|  |  | QA-5 | Using two different architecture we will achieve the portability of the system and using Draft programming language will serve moving system from one platform to another. |
|  | QA-6 |  | By using the deployment pattern, it will provide a backup server that will work if any problem happens with current servers. But still, we did not identify and demonstrate its components. |
| QA-7 |  |  | No relevant decision was made, that is necessary to identify the elements that participate in the use because that associated with the scenario. |
| QA-8 |  |  | No relevant decision was made, that is necessary to identify the elements that participate in the use because that associated with the scenario. |
|  |  | CON-1 | Using mobile reference architecture this will help achieving that system will be able to run in android, IOS software’s. |
|  | CON-2 |  | Physical structure by using balance load-balance-cluster help achieving the ability of managing many users at the same time because system will manage loads. Still, we did not define the related components that important to achieve this constrain. |
| CON-3 |  |  | No relevant decisions made. |
|  | CON-4 |  | Dividing the system into smaller portions can help with managing modifications. With multiple backup servers for the system, modifying a single portion will not affect the overall user experience. However, it is important to identify the necessary components associated with each portion before making any modifications. |
|  |  | CON-5 | Structuring the application physically using a load-balanced cluster can be beneficial in providing multiple backup servers. This can help in managing errors effectively without causing any disruption to the user experience. |
|  |  | CON-6 | Physical structure of system by defining mobile reference architectures will achieve this constrain and system will be a cross-platform. |
| CRN-1 |  |  | No relevant decision made. |
|  | CRN-2 |  | The selected technologies must meet the knowledge requirements of the development team. However, additional technologies must be chosen for communication with data sources. |
|  |  | CRN-3 | Selection of reference architectures and deployment pattern. |

### 1.2.3 Iteration 2: Identifying Structures to Support Primary Functionality.

This section outlines the outcomes of the activities performed in each step of the second iteration of the ADD design process for the Balagh system. In this iteration, we have progressed from generic and high-level functional descriptions used in the first iteration to more detailed decisions that will guide implementation and the formation of development teams. This process is inherent in the ADD method since it is not possible to design everything upfront, and disciplined decision-making is essential. In the first iteration, our objective was to establish the overall system structure, and now that goal has been achieved. In the second iteration, our objective is to consider the units of implementation, which will impact team formation, interfaces, and how development tasks can be distributed, outsourced, and executed in sprints.

#### 1.2.3.1 Step 2: Establish Iteration Goal by Selecting Drivers.

The goal of this iteration is to address the general architectural concern of identifying structures that will support the primary functionalities of the system. This identification will be useful not only for understanding how we will support the functionalities, but also in addressing CRN-1, which defines the work hours and work type. The architecture considered the primary use cases of the system, which include:

• UC-3

• UC-4

• UC-5

• UC-8

#### 1.2.3.2 Step 3: Choose One or More Elements of the system to Refine.

The element that we want to improve is located in different layers of two distinct reference architectures that we have defined. However, to support the system's functionalities, the collaboration of components associated with modules located in different layers is necessary.

#### 1.2.3.3 Step 4: Choose One or More Design Concept That Satisfy the selected Drivers.

In this iteration, several design concepts are selected, the following table summarizes the design decisions.

|  |  |
| --- | --- |
| **Design Decisions and Location** | **Rational and Assumptions** |
| Create **Domain Model** for the application | Firstly, we need to define initial domain model for our application by identifying the major entities in the domain, along with their relationships. There are no alternatives, domain must be eventually created, or it will be presented in a bad fashion. Leading to use ad hoc architecture which is hard to understand and implement. |
| Identify **Domain Object** that map the functional requirements | Each individual function should be contained within a self-contained building block, which is known as a domain object. It is possible to decompose the layers into modules without considering domain objects, but this approach increases the risk of not meeting the requirements. |
| **Decompose Domain Objects** into general and specialized Components | This pattern represents a comprehensive set of functionalities, but these functionalities are supported by finer-grained components located within the layers. These components are what we refer to as modules. Modules are associated with specific layers, such as the business logic module. |
| **Use C# and Java** mobile frameworks. | Java and C# are two of the most used languages in developing mobile applications due to their high performance and productivity. While, we connect the required reference architectures, in order to achieving the goal outlined in (QA-2). Other frameworks were not considered as the development team is already familiar with and satisfied with the performance of Java and C# (CRN-3). |

#### 1.2.3.4 Step 5: Instantiate Architectural Element, Allocate Responsibilities, and Define Interfaces

|  |  |
| --- | --- |
| **Design Decisions and Locations** | **Rationale** |
| Create only an initial domain | Entities that participate in primary use cases need to be identified and modeled but only initial domain models are created, to accelerate this phase of design. |
| Map the system use cases to domain objects | The initial identification of domain objects can be done by analysis the system’s use cases. To address (**CRN-1)** object domain are identified for all use cases in table 2. |
| Decompose the domain objects across the layers to identify layer specific modules with an explicit interface | This approach guarantees that all the functionalities defined are supported by the modules. The architecture is designed to handle tasks for the primary use cases, enabling other team members to identify the remaining modules. This helps in dividing the workload among team members. The architect has identified a concern related to CRN-5, which states that the main modules should be evaluated. This concern is specific to the main modules in the system since the modules related to the user interface are challenging to evaluate independently. |
| Connect components associated with modules using C# and Java. | These programming languages enables the control and optimization of the system's execution speed and productivity, which can be utilized in determining the critical aspects of the modules in terms of speed and execution. This will assist in addressing (CRN-5). |
| Associated frameworks with a module in the data layer | This programming language allows for the control and optimization of system execution speed and productivity. This feature can be leveraged to determine the critical aspects of modules in terms of speed and execution, which in turn can help address (CRN-5). |

#### 1.2.3.5 Step 6: Sketch Views and Record Design Decisions:

Figure 3.1 displays the system's initial domain model.

Figure 3.2 displays the domain objects instantiated for the use case model in Table 2.

Diagram

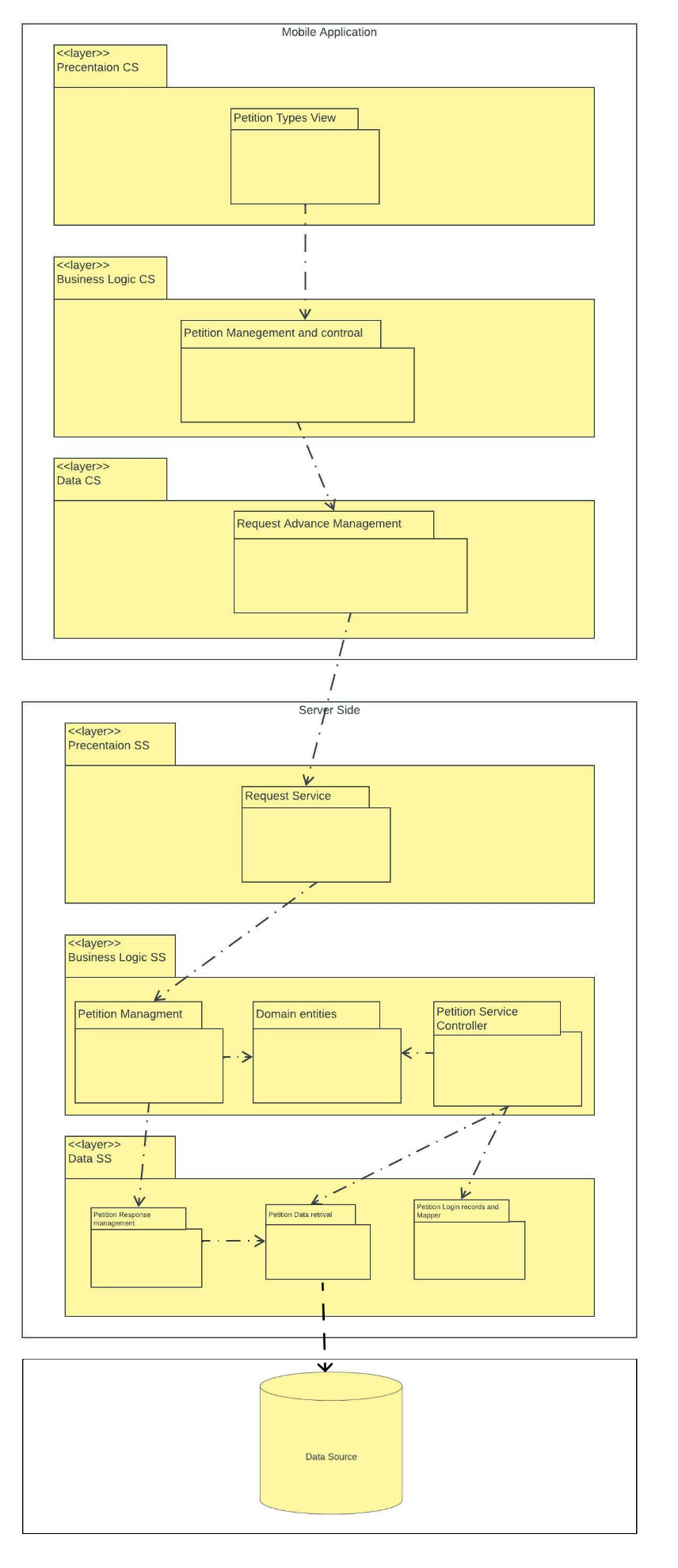
Description automatically generated

**Figure 3.1**

Diagram

Description automatically generated

**Figure 3.2**

****

|  |  |
| --- | --- |
| **Element** | **Responsibility** |
| PetitionTypeView | Responsible of viewing different types of petitions such as complaints, inquiry, and suggestions. |
| PetitionManagementAndControl | Control and management complaints, inquiry, and suggestions of the users and present it to employee ministry. |
| DomainEntities | Contains the entities from the domain model. |
| PetitionServiceControl | Their responsibility includes controlling services related to petitions, such as accepting, deleting, forwarding, and sorting. |
| PetitionResponseManagement | Managed the response of petitions related to our system by ministry employee. |
| PetitionLoginRecordesAndMapper | This component is responsible for recording and processing petitions, sending them to the database, and mapping them based on type. |
| PetitionDataRetrival | This component is responsible for retrieving data related to petitions from data sources. |
| DataSource | Responsible for saving and reserving data of petitions. |

#### 1.2.3.5.1 UC-3, UC-4, UC-5: Petition

The sequence diagram created for UC-3, UC-4, and UC-5 illustrates how petitions are managed in the system after a successful user login. The system requests data retrieval from PetitionDataRetrieval, responsible for retrieving data from the DataSource and returning it to the client for viewing petitions with the Petition class.

Timeline

Description automatically generated with medium confidence

**Mobile application sequence diagram for UC-3, UC-4, UC-5**

|  |  |
| --- | --- |
| **Method name** | **Description** |
| **Element**: PetitionTypeView | |
| *Initialize()* | Open petition representation so that user can interact with it. |
| *getPetitionRoot(): Petition* | Returns a reference to the root petition and the neighbors of this object |
| **Element**:PetitionManagementAndControl | |
| *RequestManagement()* | Request information about the petition. This method reference to the root petition data source to get the information. |
| **Element**:RequestAdvanceManagement | |
| *SendRequest(Request req)* | This method receives a request. Only this method is exposed in the service interface. This simplifies the addition of other functionality in the future without having to modify the existing service interface. |
| **Element**: RequestService | |
| *RequestPetition()* | This component is responsible for requesting the petitions from the dataSource within the server and managing the services that are associated with the petitions. |
| **Element**: PetitionManagement | |
| *RequestPetition()* | This component manages the petitions that are retrieved from the DataSource and requested by the RequestAdvanceManagement in the mobile application, and then sends them to the user's device. |
| **Element**: PetitionResponseManagement | |
| *RequestPetition()* | Manages responses from the petition data source and sends them to petition management for further analysis. |
| Element: PetitionRetriveal | |
| *RetrivePetiton(id): Petition* | Return response from its id. |

Timeline

Description automatically generated

**Sequence diagram for mobile application UC-8 (key: UML)**

|  |  |
| --- | --- |
| **Method name** | **Description** |
| **Element**: PetitionTypeView | |
| *SelectPetitionType()* | Open the petition for employees to choose type of them for managing and replaying. |
| **Element**:PetitionManagementAndControl | |
| *RequestManagement andControl()* | Request information about the petition. This method reference to the root petition data source to get the information. |
| **Element**:RequestAdvanceManagement | |
| *SendRequest(Request req)* | This method receives a request. Only this method is exposed in the service interface. This simplifies the addition of other functionality in the future without having to modify the existing service interface. |
| **Element**: RequestService | |
| *RequestPetitionInf()* | Responsible about requesting the petition information that are inside the server from the DataRetrival. Also, responsible about services that within petition. |
| **Element**: PetitionServiceControl | |
| *RequestPetition()* | Responsible for services petition that retrieved from the dataSource that requested information by the RequestAdvanceMangement in mobile application then send it to user device. |
| *AddRecord()* | Function that responsible of adding new petition record to the PetitionRecordAndMapper. |
| **Element**: PetitionDataRetriveal | |
| *RetrivePetiton(id): Petition* | Return response from its id. |
| *ViewRecordsData()* | Viewed the data that saved in data mapper and record. |

#### 1.2.3.6 Step 7: Preform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

The decisions made in this iteration helped to establish a basic understanding of how the system's functionality is supported. The architect identified the modules associated with the primary use cases, while another team member identified the modules associated with the remaining functionality. This resulted in a comprehensive list of modules to address CRN-1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Not Addressed** | **Partially Addressed** | **Completely Addressed** | **Design Decisions Made During the Iteration** |
|  |  | UC-1 | Modules across the layers and preliminary interfaces to support this use case have been identified. |
|  |  | UC-2 | Modules across the layers and preliminary interfaces to support this use case have been identified. |
|  |  | UC-6 | Modules across the layers and preliminary interfaces to support this use case have been identified. |
|  |  | UC-7 | Modules across the layers and preliminary interfaces to support this use case have been identified. |
|  |  | QA-1 | The elements that support the associated use case (All) have been identified. |
|  |  | QA-2 | Introduction to the content management performance model by identifying the load-balance cluster this will help achieving high performance of UC-8. Also, by identifying models across layers this quality attribute has been completed. |
|  | QA-6 |  | No relevant decisions made. |
|  | QA-7 |  | The elements that support the associated use case (UC-7) have been identified. |
|  |  | QA-8 | The elements that support the associated use case (UC-3, UC-4, UC-5) have been identified. |
|  |  | CON-2 | Modules responsible for collecting data have been identified |
| CON-3 |  |  | No relevant decisions made. |
|  | CON-4 |  | No relevant decisions made. |
|  |  | CRN-1 | knowledge. CRN-2 Modules associated with all the use cases have been identified and a work assignment matrix has been created (not shown). |
|  | CRN-2 |  | Additional technologies were identified and selected considering the team’s knowledge. |
|  | CRN-4 |  | The architectural concern of unit-testing modules, which was introduced in this new iteration, is partially solved through the use of an inversion of control approach to connect the components associated with the modules. |

## 1.2.4 Iteration 3: Addressing Quality Attribute Scenario Drive (QA-6)

In this section, the outcomes of each step of the third iteration of the ADD design process are presented. Based on the basic structural decisions made in the previous iterations, the focus now shifts towards assessing the satisfaction of important quality attributes. The current iteration concentrates on addressing one specific quality attribute scenario.

#### 1.2.4.1 Step 2: Establish Iteration Goal by Selecting Drivers

In this iteration, the architect prioritizes the QA-6 quality attribute scenario, which ensures that the system can function continuously without downtime by implementing backup servers. If one server fails, the other server will take over, and the failed server will restart and resolve the issue without losing any data.

#### 1.2.4.2 Step 3: Choose One or More Elements of the System to Refine

To improve the system's availability as per QA-6 scenario, the focus will be on refining the physical nodes identified in the first iteration, namely the application server and database server.

#### 1.2.4.3 Step 4: Choose One or More Design Concepts That Satisfy the Selected Driver

The design concepts used in this iteration as follows:

|  |  |
| --- | --- |
| Design Decision and Location | Rational and Assumption |
| Introduce the active **redundancy tactic** by replicating the application server and other critical components such as the database source. | By replicating critical elements, the system will be able to withstand the failure of one of the replicated elements without affecting its functionality. |
| Introduce **Petition double check** technology | Our system consists of various modules that process retrieved data to check for correctness or identify any issues. |

#### 1.2.4.4 Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The instantiation design decisions are summarized in the following table:

|  |  |
| --- | --- |
| Design Decisions and Location | Rationale |
| Deploy message when error occurs | If there are any problems in retrieving data throughout the application runs the system will show the error message follows by the error code. |
| Use active redundancy and load balancing in the application server | To ensure efficient utilization of the three active replicas of the application server, load balancing is recommended. This can be implemented using the Load-Balanced Cluster pattern. |
| Implement load balancing and redundancy using  technology support | There are various mature and easily supportable technological options available for implementing load balancing and redundancy, rather than relying on immature and complex ad hoc solutions. |

#### 1.2.4.5 Step 6: Sketch Views and Record Design Decisions

The following diagram shows a refined deployment diagram that includes the introduction of redundancy in the system.

Diagram

Description automatically generated

**Refine deployment diagram (key: UML)**

|  |  |
| --- | --- |
| **Element** | **Responsibility** |
| *LoadBalancer* | Responds to client requests (and balances their load) and dispatches them to the application servers. Clients can also get a unique IP address from the load balancer. |
| *PetitonDoubleCheck* | Ensures that data about petitions transferred between a server application and a database is accurate. |

Diagram

Description automatically generated

**Sequence diagram illustrating the messages exchanged between the physical nodes to support UC-8**

#### 1.2.4.6 Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

The design decisions made in this iteration focused on addressing QA-6, which also had an impact on QA-7. The table below summarizes the status of the different drivers and the decisions that were made during the iteration. Drivers that were fully addressed in the previous iteration have been excluded from the table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Not Addressed** | **Partially Addressed** | **Completely Addressed** | **Design Decisions Made During the Iteration** |
|  | QA-6 |  | Redundancy in the application server reduces the probability of system failure, and a passive replica is activated if the load balancer fails within the required time. However, as specific technologies for "Petition double check" have not been chosen, this driver is marked as partially addressed. |
|  | QA-7 |  | In this latest iteration, a novel issue has arisen: the management of state across replicas. As of yet, no decisive action has been taken in regard to this matter. |
| CON-3 |  |  | No relevant decisions made. |
|  | CON-4 |  | No relevant decisions made. |
|  | CRN-2 |  | Additional technologies were identified and selected considering the team’s knowledge. |
|  | CRN-5 |  | This iteration introduces a new architectural concern: managing state in replicas. There has been no relevant decision made at this point. |

# 2. Milestone Reports

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **Date** | **Status** | **Responsible** | **Issues/Comments** |
| **Initiating:**  - Business Case completed | 27 Feb | Completed | Maha and sponsor | Went very well |
| Stakeholders Register completed | 1 Mar | Completed | Maha |  |
| Project Charter completed | 7 Mar | Completed | Maha |  |
| **Planning:**  Team Contract signed | 12 Mar | Completed | Maha | Went very well |
| Software Process Model and Justification completed. | 16 Mar | Completed | Team members  stakeholders  customer |  |
| Collecting Requirements completed | 16 Mar | Completed | Maha | Issues:  some difficulties were faced in collect requirements because in the early phases the project was not clear for the team members. |
| Scope Statement completed. | 16 Mar | Completed | Maha |  |
| WBS completed. | 22 Mar | Completed | Maha | Reviewed with customer and team. |
| Project Schedule and cost completed. | 4 Apr | Completed | Maha |  |
| Risk Management Plan completed. | 19 Apr | Completed | Maha |  |
| **Executing:**  Clarification of Design Purpose completed. | 26 Apr | Completed | Fatema |  |
| Define Quality Attribute Scenarios completed. | 30 Apr | Completed | Abdullah |  |
| Define Primary Functionality completed. | 11 May | Completed | Musheera | Reviewed with customer and team. |
| List Architectural Concerns completed. | 11 May | Completed | Abdullah |  |
| Catalog Constraints completed. | 11 May | Completed | Abdullah |  |
| Review Input completed. | 28 May | Completed | Maha | Reviewed with customer |
| Establish the Iteration Goal by Selecting Drives completed. | 28 May | Completed | Maha |  |
| Choose One or More Elements of The System to Refine completed. | 28 May | Completed | Maha |  |
| Choose One or More Design Concepts That Satisfy the Selected Drivers completed. | 28 May | Completed | Noor |  |
| Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces completed. | 8 Jun | Completed | Musheera |  |
| Sketch Views and Record Design Decisions completed. | 29 Jun | Completed | Fatema |  |
| Perform Analysis of Current Design and Review Iteration Goal and Architecture of Design Purpose completed. | 29 Jun | Completed | All |  |
| Application Prototype: Interface Design and Development completed. | 3 Jul | Completed | Abdullah | Went very well |
| **Monitoring and Controlling:**  Progress Report completed. | 6 Jul | Completed | All |  |
| Milestone Report completed. | 6 Jul | Completed | Maha |  |
| Request for Proposal completed. | 6 Jul | Completed | Maha |  |
| Team Performance Assessment completed. | 6 Jul | Completed | Maha |  |
| Change Requests completed. | 16 Aug | Completed | Maha |  |
| **Closing:**  Final project presentation completed. | 3 Oct | Completed | Maha |  |
| Sponsor sign-off on project completed. | 5 Oct | Completed | Sponsor |  |
| Final project report completed. | 10 Oct | Completed | Maha |  |
| Lessons Learned completed. | 17 Oct | Completed | All |  |

* **Some strategies that helped the team succeed**
  + - Reviews at the beginning of each project phase to discuss problems and develop solutions.
    - PM align with everything and communicate always.
    - Recognized and appreciate extra effort for the team (study the team needs)

# 3. Performance Reports

**Project Name**: Balagh System

**Reporting Period:** May 11nd, 2023 to July 3th, 2023

**Work completed this reporting period:**

* Define Primary Functionality completed.
* List Architectural Concerns completed.
* Catalog Constraints completed.
* Review Input completed.
* Establish the Iteration Goal by Selecting Drives completed.
* Choose One or More Elements of The System to Refine completed.
* Choose One or More Design Concepts That Satisfy the Selected Drivers completed.
* Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces completed.
* Sketch Views and Record Design Decisions completed.
* Perform Analysis of Current Design and Review Iteration Goal and Architecture of Design Purpose completed.
* Application Prototype: Interface Design and Development completed.

**Work to complete next reporting period:**

* Monitoring and Controlling:
* Progress Report completed.
* Milestone Report completed.
* Request for Proposal completed.
* Team Performance Assessment completed.
* Change Requests completed.
* Final project presentation completed.
* Sponsor sign-off on project completed.
* Final project report completed.
* Lessons Learned completed.

**What’s going well and why:**

Overall, the development is progressing well due to the thorough planning phase we conducted. By taking the time to clearly define our requirements and establish a solid foundation for the project, we have been able to avoid significant changes or rework in the later stages. This has allowed us to stay on track with our timeline and budget and has ensured that our team is aligned and working towards the same goals.

**What’s not going well and why:**

Unfortunately, we have encountered some issues in the executing phase due to the inadequate design work. As our system is used by a diverse group of people, it is crucial that we prioritize user experience (UX) and ensure that the design is tailored to their needs. This requires a deep analysis and study of our users in order to produce a proper UX design. We recognize the importance of this phase and have determined that hiring a UX specialist designer is necessary to address these issues and ensure that our system is user-friendly and meets the needs of our stakeholders.

**Suggestions/Issues:**

* It would be helpful to have more frequent check-ins with team members to address any issues or concerns before they become major problems.
* We should consider providing additional support to the team member who experienced personal issues to ensure they are able to catch up on missed work

**Project changes:**

After careful consideration, we have identified some changes that must be made to our project scope and schedule. The addition of the new feature during the execution phase requires a high level of performance and consistency to ensure that we achieve our desired results while minimizing costs and delays. To achieve this, we will need to adjust our project plan and allocate additional resources as necessary. We understand the importance of these changes and are committed to ensuring that our project stays on track and delivers the best possible outcomes for our stakeholders.