Assignment 2: Software Architecture Document

|  |  |  |
| --- | --- | --- |
|  | Group 8 |  |
| Name | Email | Swedish Id |
| Khan, Muhammad Mohsin Qamar | muqm22@student.bth.se | 19980625T659 |
| Hassan, Syed Ali | syal22@student.bth.se | 19830417T674 |
| Shahzad,Ghilmaan | ghgh22@student.bth.se | 19950206T758 |

Ayyaz,Muawaz [muay22@student.bth.se](mailto:muay22@student.bth.se) 19960513-T771

Contents

[Design Decisions and Rationale: 4](#_Toc122269657)

[Concept: 4](#_Toc122269658)

[Structure: 5](#_Toc122269659)

[Relationship: 5](#_Toc122269660)

[Resources: 5](#_Toc122269661)

[Why the selected pattern is appropriate for the problem: 5](#_Toc122269662)

[Advantages of your choice: 5](#_Toc122269663)

[Mapping these advantages to your prioritized QAs: 6](#_Toc122269664)

[Design Decisions and Rationale: 6](#_Toc122269665)

[Glossary 8](#_Toc122269666)

[Definitions, acronyms, and abbreviations 8](#_Toc122269667)

[Part B: Modular View Documentation: 9](#_Toc122269668)

[Section 1: Primary Presentation: 9](#_Toc122269669)

[Section 2: Element Catalogue: 10](#_Toc122269670)

[2a): Elements and their properties: 10](#_Toc122269671)

[2b): Relations and their properties: 12](#_Toc122269672)

[2c): Element interfaces: 13](#_Toc122269673)

[2d): Element Behavior: 14](#_Toc122269674)

[Variability guide: 16](#_Toc122269675)

[Rationale: 16](#_Toc122269676)

[Part B: Component and Connector Documentation: 19](#_Toc122269677)

[Section 1: Primary Presentation: 19](#_Toc122269678)

[Section 2: Element Catalogue: 19](#_Toc122269679)

[2a): Elements and their properties: 19](#_Toc122269680)

[2b): Relations and their properties: 22](#_Toc122269681)

[2c): Element interfaces: 23](#_Toc122269682)

[2d): Element Behavior: 24](#_Toc122269683)

[Variability guide: 27](#_Toc122269684)

[Rationale: 28](#_Toc122269685)

**Documentation Road Map:**

This section is a given road map about all the architectural documentation. In this section we provide clear and precise information about the overall sections of the documentation what information they contain and where we can find in the document.

|  |  |
| --- | --- |
| **Section Name** | **Overview** |
| [**Design Decisions and Rationale**](#_Design_Decisions_and) | In this section all the decision that are taken for designing and architecture and their rationales are discussed. |
| [**Modular View Documentation**](#_Part_B:_Modular) | This section provides the complete overview of the Module view. |
| [**Primary Presentation**](#_Section_1:_Primary) | In this sectionUML Modular View Diagram is discussed. |
| [**Element Catalogue**](#_Section_2:_Element) | In this section each element of the system and their properties are discussed. |
| [**Element and their properties**](#_2a):_Elements_and_1) | In this section, we define all the element in modular view in the system |
| [**Relations and their properties**](#_2b):_Relations_and) | In this section the modules relationship with layers are discussed. |
| [**Element interfaces**](#_2c):_Element_interfaces:) | In this section we discuss all the architectural layers interfaces. |
| [**Element Behavior**](#_2d):_Element_Behavior:) | There are system modules behavior is discussed and how they interact with each other’s. |
| [**Variability guide**](#_Variability_guide:) | This section discussion the flexibility points in the modular view in the system |
| **[Rationale](#_Rationale:)** | This section discuss the set of logical decisions in the modular view in the system |
| [**Component and Connector Documentation**](#_Part_B:_Component) | This section provides the complete overview of theComponent and Connector view. |
| [**Primary Presentation**](#_Section_1:_Primary_1) | In this sectionUML Component and connector view Diagram is discussed. |
| [**Element Catalogue:**](#_Section_2:_Element_1) | In this section each element of the system and their properties are discussed. |
| [**Element and their properties**](#_2a):_Elements_and) | In this section, we define all the element in component and connector view in the system |
| [**Relations and their properties**](#_2b):_Relations_and_1) | In this section the modules relationship with layers are discussed. |
| [**Element interfaces**](#_2c):_Element_interfaces:_1) | In this section we discuss all the architectural layers interfaces. |
| [**Element Behavior**](#_2d):_Element_Behavior:_1) | There are system modules behavior is discussed and how they interact with each other’s. |
| [**Variability guide**](#_Variability_guide:_1) | This section discussion the flexibility points in the component and connector view in the system |
| [**Rationale**](#_Rationale:_1) | This section discuss the set of logical decisions in the modular view in the system |

**Views Overview:**

Architectural views are the overall representation of a system architecture to the meaning full stakeholders of the system. This is on the architect which view have to be develop and need for the stakeholder to understand and communicate with the system. These views enable stakeholders to verify that system address their concerns.

Architecture of any system can be represented by one or more architectural views. In our system we design two architectural views. Module view and Component-and-Connector (C&C) view.

**Module View:**

We have selected uses view from the model view because our architecture is layer-based architecture and it will be better to select this architecture because we want to see the interaction of different modules and services and how these layered are interacted. This layer architecture can define by layered architecture.

**Component-and-Connector (C&C) Views:**

Accordingly, Component &Connector of OTIS, there is an external system having LPROD, LCRM, and DHL Express. External System Connects with OTIS System having a business object Component, Data object component, UI Object component, and reports component. The UI Object connects with the Customer UI to do order tracking, Business object takes all decisions and handles the data from an external system. The data object also connects with the Business object. After a decision, it gives instructions to UI Object and UI Object connects with Reports to generate required report instructions to the system. We also have a web hosting server component and a Database server component they both interact with OTIS System and as well as interact with the Backup server/Log server component.

# Design Decisions and Rationale:

When you are going to design the architecture of any system you should take some design decisions. These decisions are taken to achieve the desire goal. Decisions are depending on the design results. we have to make architectural decision on the basis of system primary functionality and some time to achieve the desire quality level of the system. Beyond any software architecture there are several decisions are taken to design the architecture and these decisions are help full to the representations of the elements and this is called rational. Theses decision includes:

* Concept
* Structure
* Relationship
* Resources

## Concept:

When you start any design, you have several alternatives for one system. We firstly select the designing concept that states out problem and good for its solution. In our OTIS system architecture, we select the Module view concept for designing the architecture of the OTIS system.

## Structure:

After selecting the desire architectural view for the system next step is to define the structure for the view. We define a complete structure for the OTIS system architectural view as shown below.

## Relationship:

When the structure is created then we have to define the retainer ship and different connections of the views in the system. In the OTIS system we show the different kind of relationship of the layers of the views. How there are connected and share data to each other is shown.

## Resources:

All the layers in the system require the resources form the system. We allocate the require resources for the applicable layers form the different systems, peoples or any hardware. In the OTIC system we allocate the resources where its need.

## Why the selected pattern is appropriate for the problem:

As our prioritized quality attributes are performance, maintainability and availability, our solution would be Layered based system. The layered based architecture system is easy to learn as the time of the developing, testing and deployment time is 6 months. Our developers are already familiar to develop this architecture. This reduces the dependency on the modules/components on each other which increases the cohesion in the system. Cost overhead, testing is low and error handling is relatively easy.

## Advantages of your choice:

* This pattern allows for the implementation of any protocol because the protocols are concealed. Thus, I refer it as a generic model. It has the ability to adjust to numerous different regimens.
* Both connection-oriented and connectionless services are supported. So, when reliability is required, we can utilize the connection-oriented paradigm, and when we need faster transfer of data over the internet.
* Following the abstraction principle is this tiered architecture. Other layers are not significantly affected by changes to one layer.
* Compared to bundling all services into a single layer, it is easy to fix and make it available.

## Mapping these advantages to your prioritized QAs:

|  |  |
| --- | --- |
| **Decisions Benefits** | **Quality Attributes** |
| System would be available more as error fixing and testing is more easy | Availability |
| Develop the system in web based (Java) as the developer team is already familiar to the programming paradigm | Performance |
| Divide the modules into multiple components/modules as the dependency of the modules on each other is relatively low | Maintainability |
| Develop a database system backup as if database data fails there data restoration | Availability |
| Develop separate service utility to differentiate the data from multiple components to single database. | Availability |
| Layered based system provides cohesion in the system as every module/component provides its own functionality | Maintainability |

## Design Decisions and Rationale:

|  |  |
| --- | --- |
| Design Decisions | **Rationale** |
| OTIS system is hosted in a production server and multiple users that are use portal by various departments. | Due to network latency, the overhead brought on by intermediaries who handle communication, anoverall architecture approach in OTIS may have a detrimental effect on the performance of an application. To ensure that the required performance criteria are met, both the user as well as the component provider should indeed carefully develop and assess the architecture. |
| Divide domain objects into generic and specific components. | Complete functional sets are represented by application components, but these functional sets are maintained by smaller elements that are found inside the layers. In this pattern, the "components" were also what have been referring to it as components. Module specialization is related to the layers in which they are found (e.g., UI modules). There are no great options to breaking down the layers into functional modules. |
| Information can be attained by using the Mutual Database Integration technique. | The system's interactive component must do database queries. It is possible to think of the system's batch and interactive components as two separate programs (or sub - components) that share data from the same database. In this situation, the common database integration design can be applied to assist the communication between these systems. With this strategy, no modifications to the system's current components are necessary.  Discarded alternatives:   * Obtaining the data via an API, which calls for changes to the current components and would negatively affect performance. |
| Create the client application's user interface with the Java Spring boot. | The building project framework. Since the developers were already aware with it, the Java Sprint boot provide maintainability and portability.  Discarded alternatives:   * Thought was given to using the.Net Core framework, but the programmers lacked sufficient familiarity. |

# Glossary

|  |  |
| --- | --- |
| [1] | H. Cervantes, Designing Software Architectures A practical approch, Addison Wesley, 2016. |
| [2] | L. B. P. Clements, Software Architecture in Practice, 2004. |
| [3] | F. Bachmann, "Managing Variability in Software Architectures". |
| [4] | F. Bachmann, "Documenting Software Architecture: Documenting Behavior," Architecture Tradeoff Analysis Initiative, 2002. |
| [5] | F. Bachmann, "Documenting Software Architecture: Documenting Interfaces," Architecture Tradeoff Analysis Initiative, 2002. |

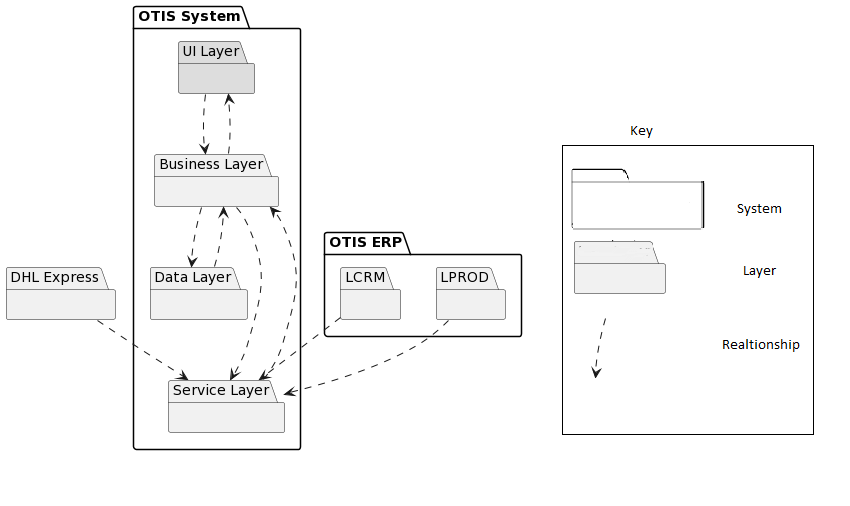
# Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| Term | Meaning |
| Lycia | Lycia Company |
| LRPO | LPROD Lycia Production System |
| LCRM | Lycia Customer relationship management system |
| HTTP/HTTPS | Hyper text transfer protocol/ secure hypertext transfer protocol |
| ORM | Object Relational Mapping |
| API | Application programming interface |
| UI | User Interface |
| QA | Quality Attribute |
| UC | Use Case |

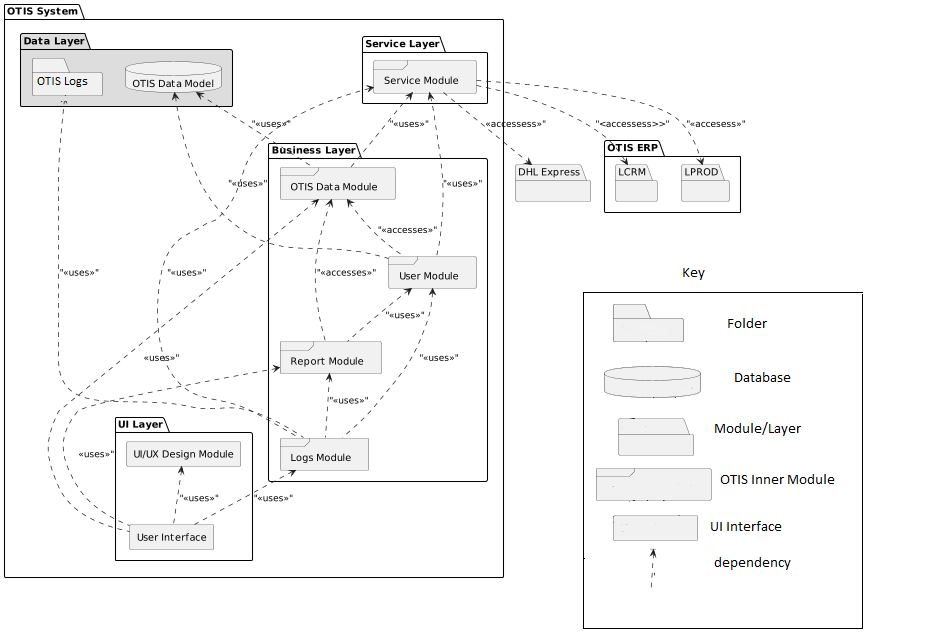
# Part B: Modular View Documentation:

## Section 1: Primary Presentation:

* **UML Modular View Diagram**



* **Uses view in Modular View**



## Section 2: Element Catalogue:

### 2a): Elements and their properties:

Our system would consist of four layers in which the data is managed. The layers are listed below:

* OTIS Service
* Business Layer
* OTIS Data Model
* UI Layer

**OTIS Service:**

OTIS service consists of the module which has the responsibility to fetch data from the external system and provide data to the OTIS modules in the business layer. The layer consists of one module listed below:

|  |  |
| --- | --- |
| Element | Responsibility |
| Service Module | The Module would fetch data from the external system through required requests and provide data to the business layer of our System. When the user logs into the system, the system would authorize the user by getting the user details from the LCRM. The service module consists of triggers that automatically dump new data from the external systems to the OTIS system |

**Business Layer:**

Business layer of our system consist of module which consists OTIS changes in the system. Data storing, Data fetching, Data retrieval, system logs logic is written in the layer. The business layer consists of following modules:

|  |  |
| --- | --- |
| Element | Responsibility |
| Report Module | Consist of all different types of reports which user could download and view from the system. System can filter through parameters and give different format to the authorize user. |
| Logs Module | The logs are categorized into system logs, audit logs and event logs. Module get data from service, reports and user module to get all the events with timestamps in log file in the server. |
| OTIS Module | Get data from service module and store data in the OTIS database. It’s also provide the OTIS modules data which is needed by the user. |
| User Module | Stores data of different users and their authorization level in the system. Stores and retrieves data from the database and matches the authentication system by validating the given data. User access is also managed in the module |

**OTIS Data Model:**

Data Model consist of OTIS relation database and Otis Log files. The data model stores the incoming data from the modules and perform action through trigger, views queries.

|  |  |
| --- | --- |
| Element | Responsibility |
| OTIS Logs | Consist of folder of different logs of the system. OTIS Logs interact with the logs module and create edit logs files in the system. The logs data is also needed by authorized users to check the activity, error, event in the system |
| OTIS Database | Consist of Relational Database which stores and saves data. Data views, indexes are created in the database in order to get better performance |

**UI-Layer:**

UI Layer consists an interactive dashboard where users can see all their action they can perform. It consists of 2 elements

|  |  |
| --- | --- |
| Element | Responsibility |
| UI/UX module | UI/UX module consist of UI elements which is an open-source component used for user interface. It is used by the user interface |
| User Interface | User Interface is screens where users can interact with the system and perform actions. Each user type have different User interface |

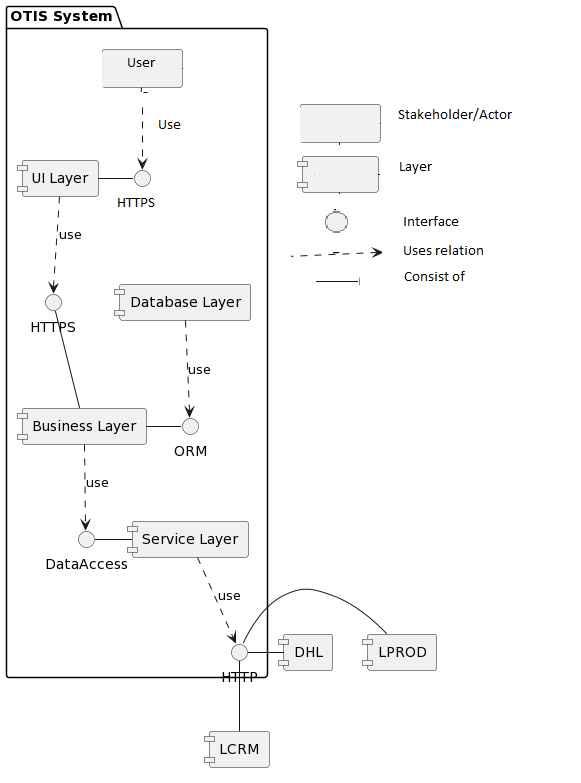
### 2b): Relations and their properties:

|  |  |  |
| --- | --- | --- |
| Element | Relationship | Responsibility |
| Service Module | User Module uses Service Module | User module get data user data from Service Module. A automated call would generate from Service Module which get the data from external system |
| Logs Module uses Service Module | Each time Service Module dumps the data into the system, a log is being entered into the system |
| OTIS Module uses Service Module | External Systems (LPROD, DHL) is maintained in our system. Service module dumped data every 15 minutes |
| OTIS Data Model | OTIS Data Module uses OTIS Data Model | OTIS Data Module saves LCRM, LPROD, DHL Express data into OTIS Database through OTIS Data Model. New Data after 15 minute replaces the old data in the database |
| User module uses OTIS Data Model | After getting user data from LCRM, User Module authorizes and saves the data through OTIS Data Model. This OTIS User’s List is replaced after every week.  User module authorizes by giving authority by checking the user data from OTIS Database |
| OTIS Data Module | Report Module uses OTIS Data Module | Report Module get data from OTIS Module of LCRM, LPROD, DHL and generate a report according to provided Data |
| User Module uses OTIS Data Module | User Module get the LCRM, LRPOD and DHL data from OTIS Data Module and provides user with the data. |
| UI Dashboard Uses OTIS Data Module | UI checks the list of orders and their progress details through OTIS Data Module |
| Report Module | UI Dashboard uses Report Module | Authorized users from UI screen can generate report by providing the parameter and order list in the system |
| Logs Module uses Report Module | Every Report generate request by the user, log module create a log in the system |
| User Module | Log Module uses User Module | Every Time User request a login and view Data of their orders and track orders, a log through log module is created |
| Report Module uses User Module | User module uses User details from User Module and generates report |
| OTIS Logs | Logs Module Uses OTIS Logs | All OTIS logs from the modules are saved in OTIS Logs folder in server through Logs Module |
| UI Module | UI Dashboard uses UI Module | External styling components are used in Dashboard for good user experience. |
| Logs Module | UI Dashboard uses UI Module | System Admin checks the logs of the system and can check errors with timestamps |

### 2c): Element interfaces:

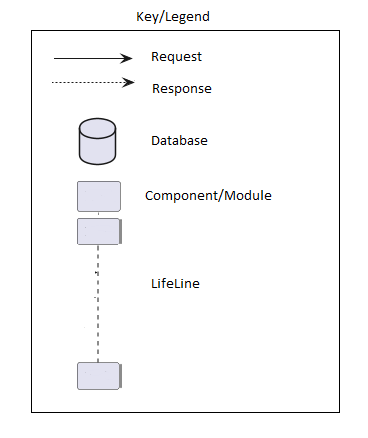
Our system is based on layered architecture and layers can different interfaces which talk with each other. Our system is Web Based and layers interact with each other using the HTTPS web as shown in figure. Service layer uses the HTTPS (JSON, XML) API of external system to fetch data which serves as a utility in our system. Our Business layer takes the data and deliver it into respective database using the ORM (Object Relationship Model). System users can view the data through a web browser by going to the provided route.

* Interface Diagram



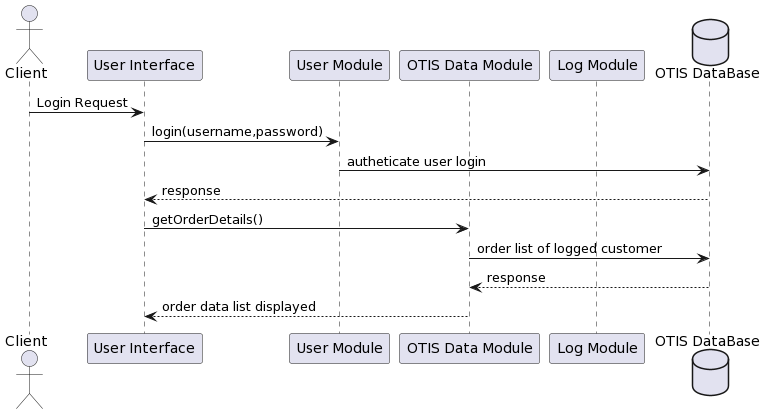
### 2d): Element Behavior:

Our application interacts with modules and database in order to handle the request from the user of the system. Our system is updated using an automated service through Service module. Service modules fetches data from external system and replaces the existing data from database through respective module functions. The behavior of three primary use cases of our system is shown through sequence diagram below:

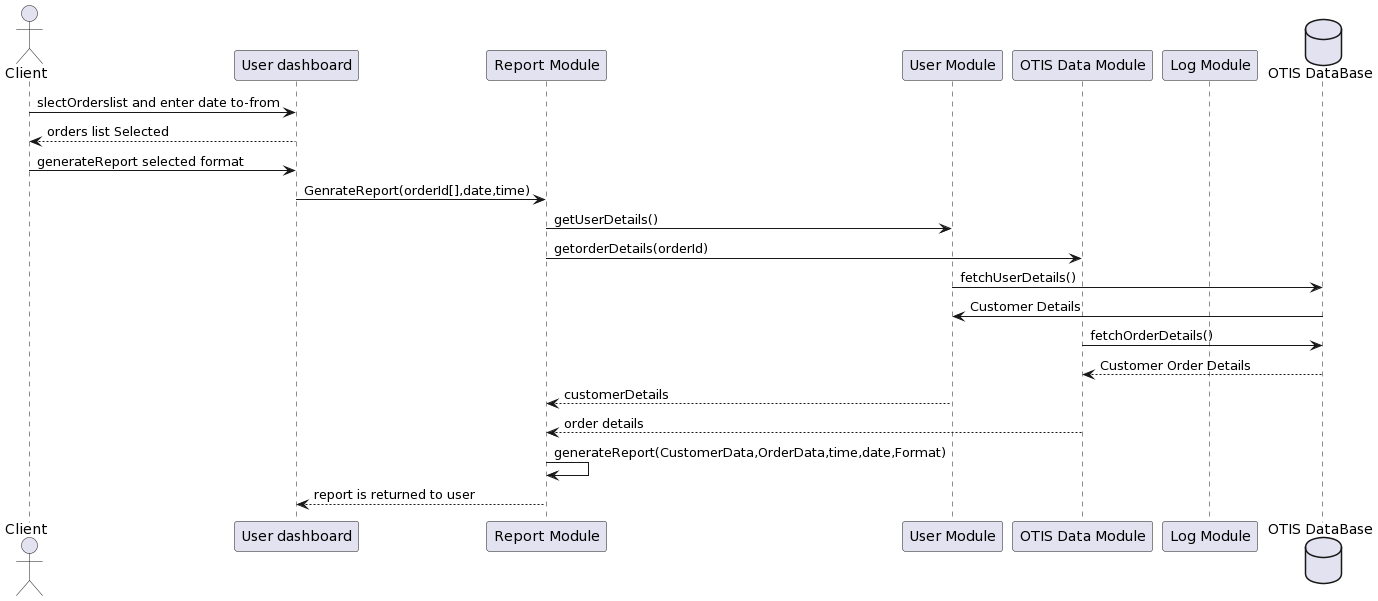


#### Sequence Diagrams:

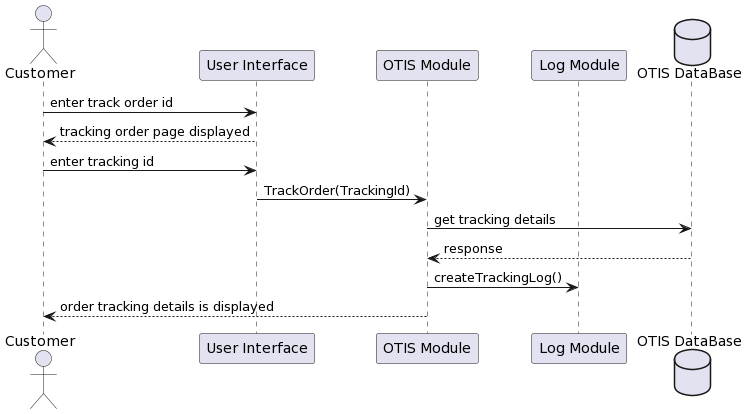
**Use Case 1: Manage Customer Order Record**



**Use Case 2: Generate Order Report**



**Use Case 3: Track Order Record**



## Variability guide:

Lycia has more than 400 client spread all across Europe. As our data is extract from external system LCRM, LPROD and DHL express, our service module triggers an API call to the external system and get relative data and saves it into OTIS database.

|  |  |  |  |
| --- | --- | --- | --- |
| SR No. | Variability | Module | Rationale |
| 1 | An API call is generated to the relevant external system if data isn’t present in our data is required by the user | Service Module | If data isn’t present in the system, OTIS system needs to validate that the data is in sync with the external system data |
| 2 | If the external system data dump fails, the system provides the latest data from database | OTIS Module, User Module | The system needs to displays user with the latest possible data if the external data dump causes an error. |
| 3 | If OTIS database fails, backup database is also maintained and used in operation | OTIS Database | If the database crashes/destroys accidently, the user order information must be backed up |
| 4 | UI/UX Module can be changed and external UI library can be used. All stakeholders have different UI to interact with the system | UI Layer | Using external UI library will help our developer team to implement as they are familiar with the UI components. |
| 5 | If the user order data is not updated latest to 30 min, system also provides user data with label “Legacy data” | OTIS Module | Will let the user know if the data is in-sync with the latest external system data. If the user queries and the system doesn’t present the same thing, user needs to get notified |

## Rationale:

#### Architectural Drivers:

|  |  |  |
| --- | --- | --- |
| Design Decision | Addresses | Rationale |
| Layered-oriented architecture: Separate layers in the system | QA2: Maintainability | As our main three major quality attributes are Performance, Maintainability and Availability. Given the conditions it will be easy to develop and maintain. As dependency is low, it addresses availability. Performance will be good as system would have less than 10000 users. |
| Using Web Interface | Concerns C5 | Delivery time is 6 months. Our team is trained to build Web application |
| Create Separate Report Module in the system | UC2: Generate Order Report | Report record and report UI is created separately from the system. It increases cohesion and it would be easy to keep track of the reports generated and log management |
| Creation of a separate database | QA3: Availability | If the data is not provided by the external system, our system would fetch the latest data from database |
| Java Spring boot is used for selected Tech-stack | Constrain C2 | Our team had experience in developing the system in Java spring boot. As we need to integrate the OTIS system with the external systems and system needs to be up and running in 6 months. |
| Logs management is done through a module and saved separately in the server as log files | UC6, UC5 | Logs Event, System and audit logs are saved and handled through a separate module and saved in log folder in server |
| Order detail and order tracking are fetched through the service and saved in the database through OTIS Module | UC3: Track Order Record | As external Systems (DHL, LPROD) provide us the data through JSON, the data is dumped every 30 minutes into the system. A database query would fetch the data to user. If the data is not present in the OTIS database, An API call would check in the external system and display the data if present. |
| If External system data dumping fails, the error is logged into system logs and the system admin is notified through email | QA1 : Performance | The data dumping fail meaning that our system has the data but the data is not updated with the system. So an Error message and notification needed to be generated in order to alert the system admin to fix the issue |
| Designed in layered Modular Architecture | QA2: Maintainability | As Performance and maintainability were our main Quality attributes, We decided to opt for an layered architecture |
| Increase Cohesion by separating modules which have different responsibility | QA2: Maintainability | Separating modules will increase cohesion in the system and it would be easy to test and maintain |
| Create Separate Module for Logs of all the system | Concern C9 | As the system have multiple layers and it would be easy to track the logs. If a system fails, our system logs are not affected |

#### Tactics: (Change tactics with the performance)

We as a team looked into the quality attributes, use cases, constraints and concerns implemented

|  |  |  |
| --- | --- | --- |
| Tactics | Tactics QA | Module Implemented |
| Create a database backup and update it weekly | Performance(Maintain multiple copies of Data) | Data Model(Data Layer) |
| Trigger data dump of users from LCRM once every 4 weeks | Performance(Manage Event Rate) | Service Module |
| Expectation handling is done if an error occurs in the generation of reports and alert is notified to System Admin | Maintainability(Fault Detection) | Report Module |
| If the database doesn’t have the required order tracking data, an API call is generated to the external system LPROD,DHL | Availability (State Resynchronization) | OTIS Module |
| System Administrator can view the error logs along with the timestamps and module | Availability (Expectation Detection) | Logs Module |
| Service module and Database server will be monitored and how well the system is performing | Availability (Conditional Monitoring) | Service Module, Database Server |
| Modules is split to increase the cohesion in the system | Maintainability (Split Module) | Business Layer |

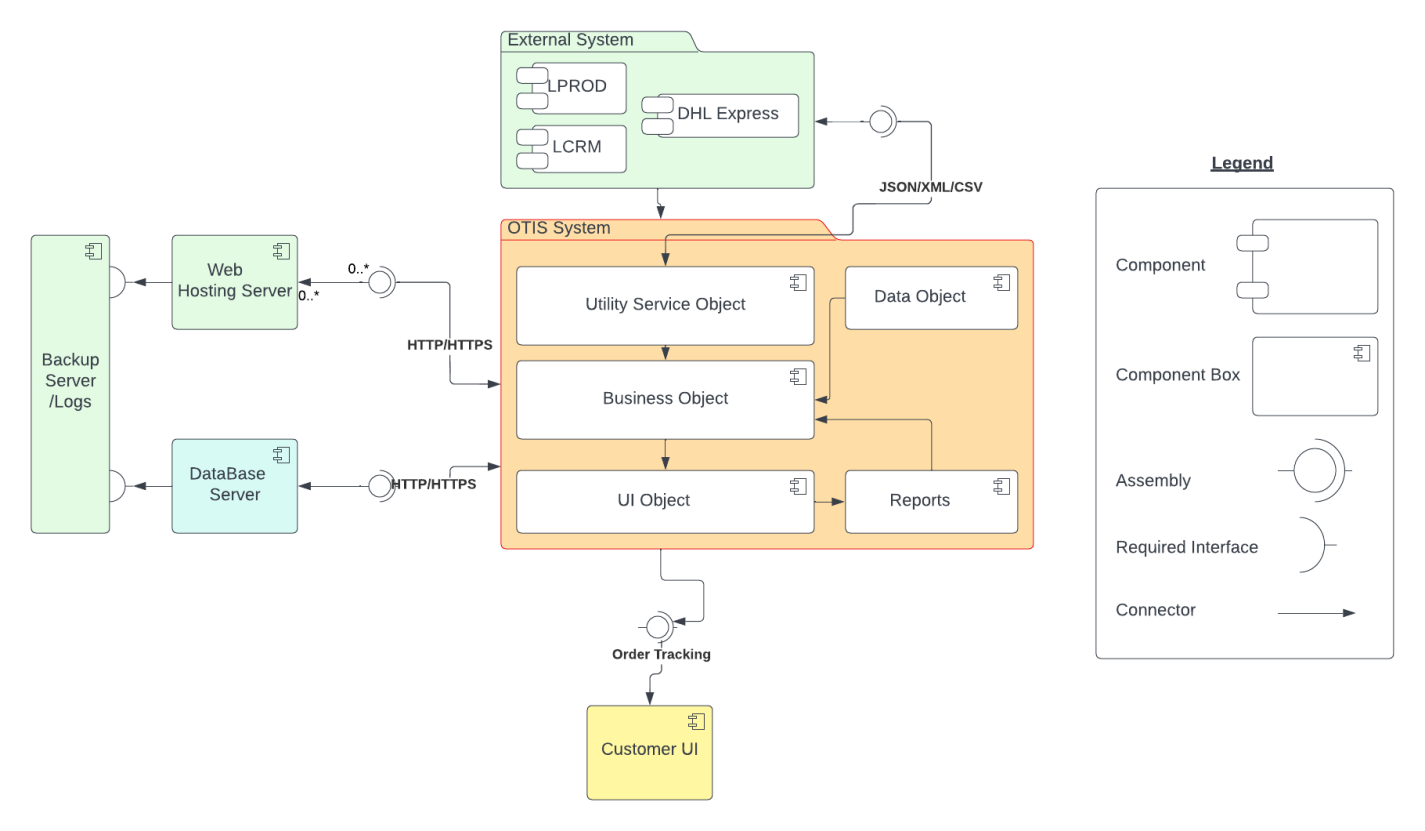
#### Other Architectural Decisions:

|  |  |
| --- | --- |
| Decisions | Rationale |
| UI/UX module is used by the user interface in importing UI components | As there are multiple stakeholders in the system, there are multiple dashboard for the system. We use pre-build UI/UX components. |

## Part B: Component and Connector Documentation:

### Section 1: Primary Presentation:

* **UML Component and Connector Diagram**



## Section 2: Element Catalogue:

### 2a): Elements and their properties:

Our OTIS system would consist of multiple components in which the data is interact with each other at run time via component and connector structure. The C&C are listed below:

* OTIS System Component
  + Utility Service Object
  + Business Object
  + UI Object
  + Data Object
  + Reports
* External system Component
  + DHL Express
  + LCRM
  + LPROD
* Web Hosting Component
  + /Tomcat /Apache/IBM Web Sphere
* Database Server Component
  + My SQL
* Backup Server /Log Server component
  + Log Server
* Customer UI Component
  + Customer UI

**OTIS System Component:**

The main component and connector structure is OTIS System (Component) which consist of five different component interact with each other and depend on every modules in OTIS main component. It’s also interacting with external system, databases component and web hosting component. Below are details description/ responsibility of each component.

|  |  |
| --- | --- |
| Component | Responsibility |
| Utility Service Object | This service utility work continuously extracting the data from external components. The utility service object fetches data from the external system through requests and provides data to the business layer of our System and data stored on database. When the user logs into the system, the system would authorize the user by getting the user details from the database updated user profile via LCRM user lists. The service module consists of service constantly working as background service for extracting data. |
| Business Object | Business Object (Component) of our system consist of multiple interaction component which consists OTIS all business object / business rule has been written in the system. Data storing, Data fetching, Data retrieval, system logs logic is written in the component. This component has main responsibility to connect all other components like data module, reports, service object and UI objects. |
| UI Object | UI Objectcomponent consist of UI elements which is an open-source component used for user interface. It is used by the user after login to the system by authentication methods. Its consist of multi level UI/UX with fully responsive web pages. Has main dashboard as landing page where user can view partial views to look many other UI of components. |
| DataObject | The data object component is consist has many classes for data structure used by ORM (object relational model). Also has many other custom model used by business object component. |
| Report | The report component consists of all different types of reports. Has separate module for users so user could download and view from the system. System can filter data through parameters and give different format to the authorize user. |

**External Services Component:**

External system componenthandle all component external component to interact and provide ETL (extract, transform, load)/data as (JSON/XML) from the external system (component) to the utility service object in OTIS system where the data has stored in database. The utility component consists of three module listed below:

|  |  |
| --- | --- |
| Component | Responsibility |
| External Services Component | This external services component work continuously extracting the data from external components like LCRM, DHL Express, and LPROD. The utility service object fetches data from the external system through requests and provides data to the utility service object and the data stored on database. When the user logs into the system, the system would authorize the user by getting the user details from the database updated user profile via LCRM user lists. The external service module consists of scheduler service constantly working as background service for extracting data. |

**Web Hosting Component:**

The web hosting component of OTIS system deployed on web hosting component on server. Used to manage and handling request from OTIS main component. Handle web load balancing, HTTP / HTTPS certifications etc.

|  |  |
| --- | --- |
| Element | Responsibility |
| Web Hosting Component | The web hosting component responsible for web application web hosting via web hosting services. It can manage http and https certificates and different security certification for web hosting. Its work locally of Lycia environment but also on public access by public IPs. |

**Database Server Component:**

The database server component of OTIS system consists of My SQL. The server has capability to store large amount of data with backup facility to stored data on backup server.

|  |  |
| --- | --- |
| Element | Responsibility |
| Database Server component | The database server component responsible for storing data in relational database. It can manage and store large amount of external system data, user information, and product tracking information. Also has capability to stored file data on db server.  . |

**Backup Server Component:**

The backup server component of OTIS system consists of db server and file server for storing and backup database files in case of failure of system and in any component failure. The server has also stored log of system. The backup server has capability to store large amount of data.

|  |  |
| --- | --- |
| Element | Responsibility |
| Backup Server Component | The backup server component responsible for backup data via RAID (redundant array of independent disks) used to storing data either in system logs and database backup. It can manage and store large amount of external system data, user information, and product tracking information. The backup server component has log services for store event logs / user exception logs and other logs. |

**Customer UI Component:**

The customer UI component consists of interactive dashboard where users can see all their action they can perform. Handle lots of view for tracking order getting daily reports and extract details of order products.

|  |  |
| --- | --- |
| Element | Responsibility |
| Customer UI Component | The customer UI component module consists of UI elements which is aweb based interactive UI used for user interaction. It is used after user authenticated by system and user see their screens and performs actions. Different company has various type of screens according to requirement assign by system admin team. |

### 2b): Relations and their properties:

|  |  |  |
| --- | --- | --- |
| **Element** | **Relationship** | **Responsibility** |
| Business object Component | External System Component | The business object component get data from external system component .A automated utility service extract data and generate ETL from utility component. |
| UI object | Every time when User request from UI object the system would be get data from database and provide data in UI response. |
| Reports | The reports module used to generate report. As system extract data from external system and generate reports via different filter applied on system. |
| Data object component | Data object component uses business object component | The data object component interacts with business objects via ORM for mapping of database tables also support custom classes for mapping multiple tables. |
| Utility service component | Business object component | The component use to interact with business logic written in Business object to store data on database server |
|  | External service component | The external service component used to extract data services from LCRM , DHL Express, LPROD and push to utility service component for data storing on database |
| Report | UI Object component uses Report Module | After authentication users from customer UI screen/ UI interface can generate report by providing the parameter and track order reports in the system. |
| Logs Module uses Report Module | Every Report generate request by the user, log module create a log in the system |
| UI Object Component | Customer UI Component | From customer UI component user request to login and view Data of their orders and track orders, a log through log module. |
| Report component uses User UI component | User module uses User details from User Module and generates report provide reports to other users .download reports in various format like pdf and xls. |
| Logs component | Logs Module Uses business object component | All OTIS logs from the modules are saved via business logic written in OTIS database and share backup folder on backup server |
| External system component | External system component uses utility services | External system component has services for fetch data from LCRM, LPROD, and DHL express to push in utility service component from storing data on database. |
| Web hosting component | Web hosting component uses main OTIS component | The web hosting component has interacted with OTIS main Component hosting via IIS. |
| Database server component | Database server component uses main OTIS component | The database server component used to save data from business object component (OTIS main component). |
| Backup server Component | Backup server component uses database / web hosting server component | The backup server component uses database and web hosting server component for backup database and log files. |

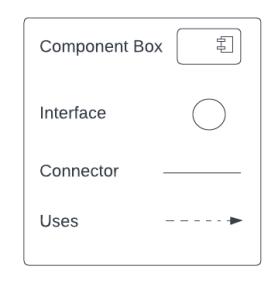
### 2c): Element interfaces:

OTIS is based on Layered based architecture but has services to interact with each component. The interface is mainly interacting with other components of system. There are five interfaces for the system. The system would develop from local machine of developers to a staging site and finally integrated to production site on production web hosting server. There are below component worked with each other for interaction of system

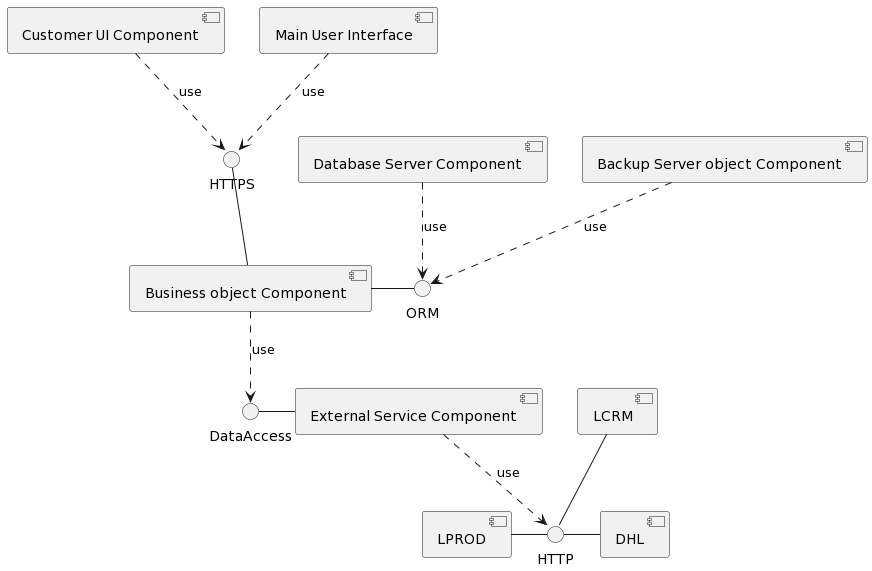
1. External System
2. Main Dashboard User Interface
3. Customer UI
4. Web Server
5. DB Server
6. Backup Server

System has many interfaces depending upon the authorization level given to each module and user respectively. Every interface depending on other required interface. Below diagram showing how interface work on OTIS system.

* **Interface diagram**
  + Key / Legend



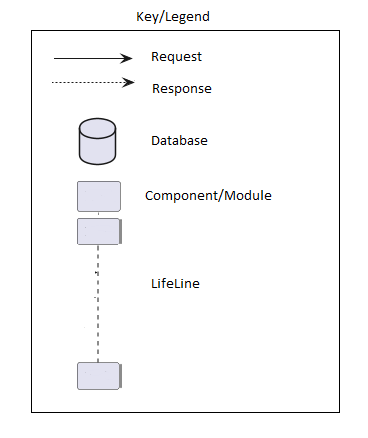
* + Diagram



### 2d): Element Behavior:

The OTIS application interacts with each component via user interface and business object component in order to handle the request from the user of the system. The system shall extract the data from external services component and then stored on database server. The external system utility, as scheduler services that trigger on every 30 minutes to update the database according to new data available from system. The system shall using web hooks on LCRM and LPROD system to update regularly. The behavior of 3 primary use cases of our system is shown through sequence diagram below:

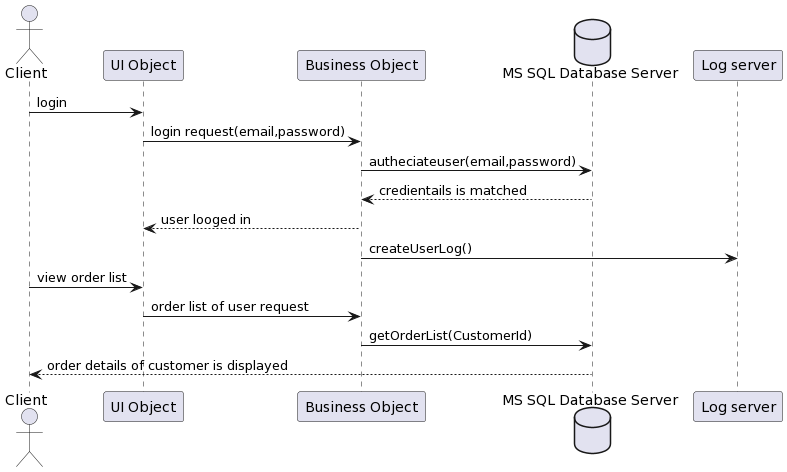
Key/ Legend:



#### Sequence Diagrams:

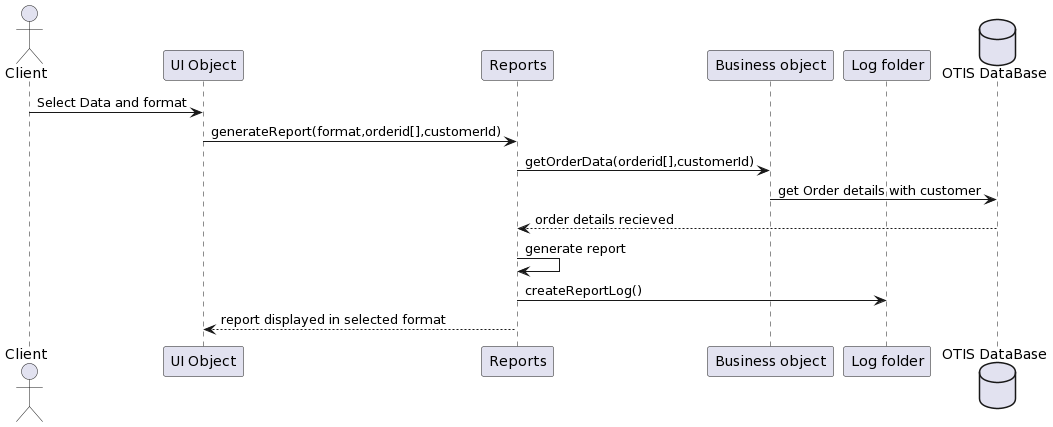
Use Case 1:

* UC1-Manage Customer Order Record



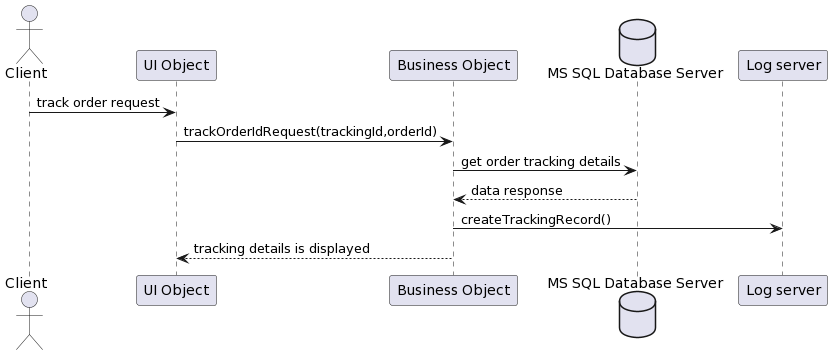
#### Use Case 2:

#### UC2-Generate Order Report



#### Use Case 3:

UC3-Track Order Record



## Variability guide:

The system shall design as a large scale system which has component approach so when a component needed we use to connect a component on business object component as the system has been design to extract the data from any external system either new module is coming that directly integrated to external system and create business objects on OTIS system. It is relatively good approach for getting integrated any module to the system in future either its web service or any other data services.

|  |  |  |  |
| --- | --- | --- | --- |
| SR No. | Variability | Module | Rationale |
| 1 | An Hook call trigger on every 2 hour for getting data from LPROD,DHL Express and LCRM objects | External Service Component | If data call fails to get data or any service failure then data is validate again for another call if error found persistently then system should trigger to write log on log file server. on other case if no new data is not found then system should not trigger to update anything on database |
| 2 | If the system data dump failure then the system display previous data only depend upon last update from last database | OTIS main Component Module, Business object Component | The system shall always show the latest updated data whenever new data updated. In case of any backend services failure the system shall display last updated records not affected on system. |
| 3 | In case of OTIS database service failure then backup RAID Service provide service with latest database backup. | OTIS Database Component | The system shall working normally incase of production database crash or service abundant. backup service start on immediate basis. |
| 4 | The Customer UI component/pages is capable to assign according to roles and rights. | UI Object Component | The admin will changes and assign different pages according to role and rights assign to customer UI. |
| 5 | If customer track an order but data is not found or upload on server so default massage has been display on notification menu | OTIS main Component | The system shall let the user about the sync data via default response message via response mapping table |

## Rationale:

#### Architectural Drivers:

|  |  |  |
| --- | --- | --- |
| Design Decision and location | Addresses | Rationale |
| Select separate service component for external system | Quality Attribute (Maintainability) QA1 | The system shall cover maintainability quality attributes so separate external component is easy to build and deploy as service. If any other new service need to attached the only process will be added as factory pattern implemented so it’s easy to maintain and component efficiency is very high so not dependence on main OTIS system. The background service constantly run and data extract and upload to system with 10 min time span. |
| Select Responsive web Interface on OTIS System UI Component | Concerns | The system shall build with responsive design so it’s easy to use and response on any web browser and mobile devices. The quality of web responsive interface is light weight and scalable. |
| Select separate Reports Component | UC2 | The system shall build with report component separately because development and maintenance is easy and other 3rd party libraries like DevExpress and Telerik tools charts/ pivot grid easily integrated with the reports component |
| Select database component with capability on separate backup DB/Log server | Quality Attribute Availability | The system shall use separate database component. We setup separate backup facility so system should be available 24/7 in case of any disaster the system shall available by backup service instance. |
| Select web hosting component on web server | Quality Attribute Availability | The system shall use separate Webhosting component rather than with database separate hosting facility should be available in case of any failure on system from backup hosting server. |
| Use External system component batches for data receiving. | Constrain | As per our finalize decision we use following format to receive ETL from external system as get from web service (JSON/XML) and file format is (CSV, XML). |
| Server logs component on backup server | UC6 | Each log either system log / application log and event log is stored as text file by time stamp on share log folder. So it’s easy to access by developer and support team. |
|  |  |  |
| In case of system failure | QA 3 Performance | The system should be capability to switch to backup application server if any failure of component. The automated load balancer used to redirect system to replica backup production web server and db server. So system should be available 24/7. |
| Each component separately work with their respective sub component | QA 1 Maintainability | OTIS system work with sub component (UI component, Business Object component ,Data Object ,Reports component) and external system and db/web server component work separately so I case of any new module injection the system should be not depended on other system. |
| Loosely couple system and implementing approach | QA 1 Maintainability | System shall be loosely couple with other component and more generic so integration is easy and implementation approach is quite easy according to SOLID principle. |
| System upgrade and maintenance downtime | Concern | The system shall take down time every time at night 2 hours for system upgrading and maintenance |

#### Tactics:

In designing for OTIS system components and connectors we deeply look into QA, use cases, constraining and implementation approaches.

|  |  |  |
| --- | --- | --- |
| Tactics | Tactics QA | Module Implemented |
| Log management on event of success and failure. | Availability | OTIS Business Object Component |
| UI assign for managing user and assign screens to customers | Performance | UI Component |
| Check records to dump data to db server on every 10 min | Performance | Utility Service Component |
| Local and global exception handling | Maintainability(Fault Detection) | Business Object Component |
| External API Expose for HTTP/HTTPS web services | Availability | External Service component |

#### Other Architectural Decisions:

|  |  |
| --- | --- |
| Decisions | Rationale |
| Customer UI via UI Object Component | As we build UI Object component for customer so the admin would be assign the custom UI and roles and rights for customer for view reports and tracking the order. |