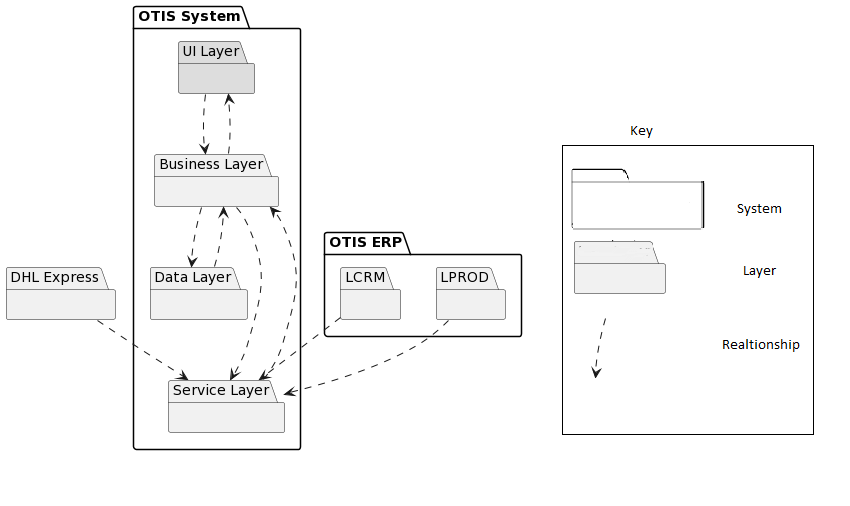
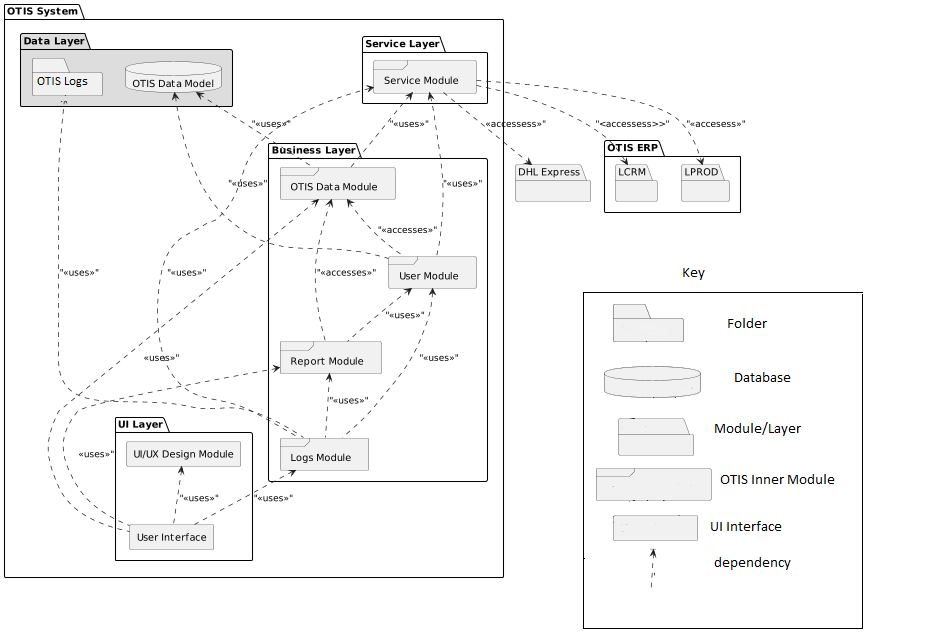
# Part B: View Documentation:

## Section 1: Primary Presentation:

UML Modular View Diagram





## Section 2: Element Catalogue:

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

Our system would consist of 4 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:

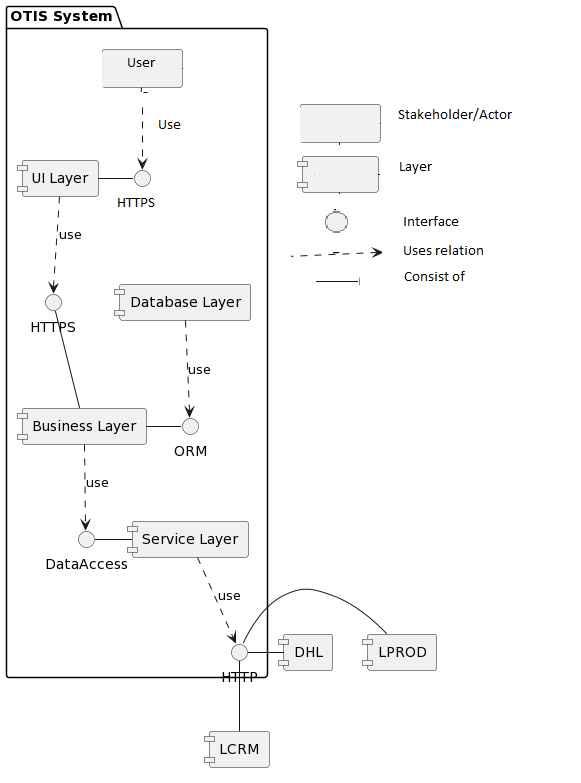
|  |  |  |
| --- | --- | --- |
| Element | Relationship | Responsibility |
| Service Module | User Module uses Service Module | User module get data user data from Service Module. A automated call would generate from Service Module which get the data from external system |
| Logs Module uses Service 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:

There are 3 interfaces for the system for each stakeholder. The system would develop from local machine of developers to a staging site and finally integrated to production site. The 3 main stakeholders of the systems are:

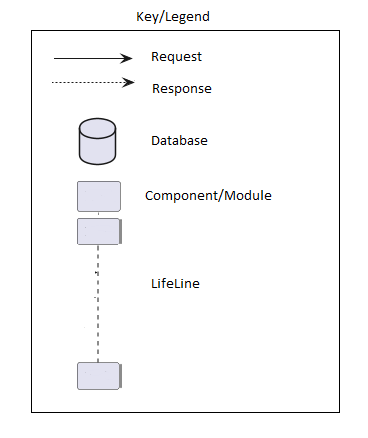
* Customers
* Key Account Managers(KAM)
* System Administrators

System has many interfaces depending upon the authorization level given to each module and user respectively. The interfaces diagram is shown below:



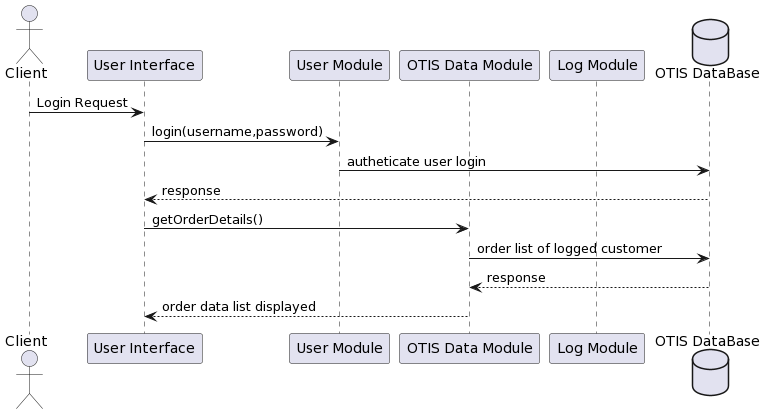
### 2d): Element Behavior:

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 3 primary use cases of our system is shown through sequence diagram show below:

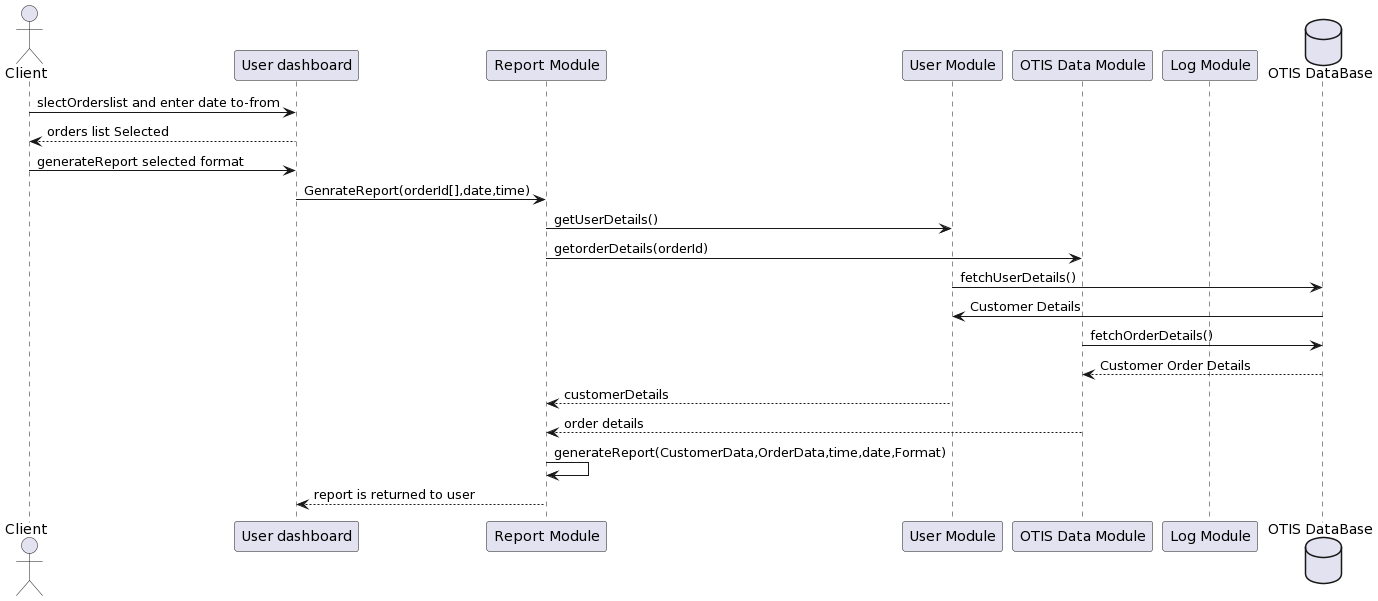


#### Sequence Diagrams:

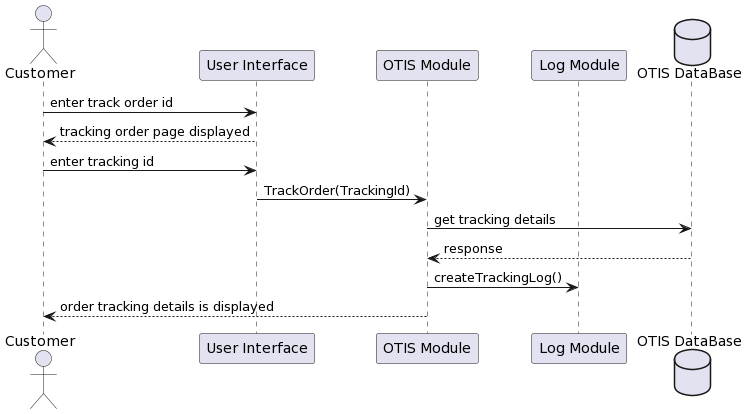
**Use Case 1: Manage Customer Order Record**



**Use Case 2: Generate Order Report**



**Use Case 3: Track Order Record**



## Variability guide:

Lysia has more than 400 client spread all across Europe. As our data is coming 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 which is required | 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 our 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 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 3 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 for now |
| Using Web Interface | Concerns C5 | Delivery time is 6 months. Our team is trained to build Web Apps |
| 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 Data Base 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 pre-build UI/UX components |