Masterize you Degree

Submitted by

- Erfan Davoodi - 4842444
- Ehsan Eslami Ziraki - 4850649
- Sahitya Reddy Bollavaram - 4849759

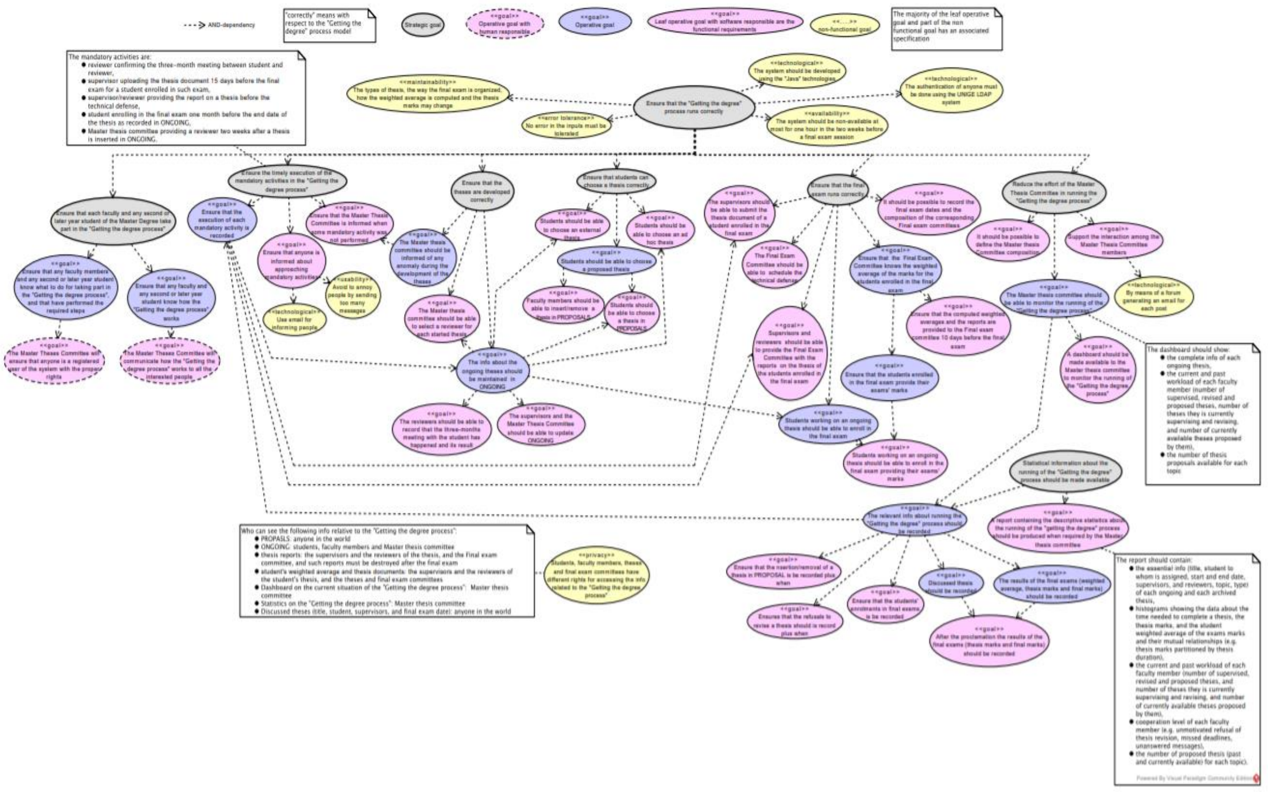
This document shows the design for “Masterize your degree” which is a greenfield system done using the ADD 3.0 method. This system details an initial design round composed of three iterations. The following shows the business content, and then we summarize the requirements for the system.

Design Purpose:

The main aim of Masterize your degree is to provide a platform for all the students of University of Genoa’s Master course, Supervisors, Reviewers, and other faculty members involved in the process of getting a degree. The system facilitates various activities such as documenting and storing information, providing notices about deadlines, meetings organization and other such activities involved in getting a degree. These are discussed in detail in the further steps.

System Requirements:

The diagram below shows how all the requirements are derived from their motivations.



AND-dependency
Strategic goal
<<goal>> Operative goal with human responsible
<<goal>> Operative goal
<<goal>> Leaf operative goal with software responsible are the functional requirements
<<...>> non-functional goal
Ensure that the "Getting the degree" process runs correctly
<<maintainability>> The types of thesis, the way the final exam is organized, how the weighted average is computed and the thesis marks may change
<<error tolerance>> No error in the inputs must be tolerated
<<technological>> The system should be developed using the "Java" technologies
<<technological>> The authentication of anyone must be done using the UNIGE LDAP system
<<availability>> The system should be non-available at most for one hour in the two weeks before a final exam session
Ensure that each faculty and any second or later year student of the Master Degree take part in the "Getting the degree process"
<<goal>> Ensure that any faculty members and any second or later year student know what to do for taking part in the "Getting the degree process", and that have performed the required steps
<<goal>> Ensure that any faculty and any second or later year student know how the "Getting the degree process" works
<<goal>> The Master Theses Committee will ensure that anyone is a registered user of the system with the proper rights
<<goal>> The Master Theses Committee will communicate how the "Getting the degree process" works to all the interested people
Ensure the timely execution of the mandatory activities in the "Getting the degree process"
<<goal>> Ensure that the execution of each mandatory activity is recorded
<<goal>> Ensure that the Master Thesis Committee is informed when some mandatory activity was not performed
<<goal>> Ensure that anyone is informed about approaching mandatory activities
<<usability>> Avoid to annoy people by sending too many messages
<<technological>> Use email for informing people
Ensure that the theses are developed correctly
<<goal>> The Master thesis committee should be informed of any anomaly during the development of the theses
<<goal>> The Master thesis committee should be able to select a reviewer for each started thesis
<<goal>> The info about the ongoing theses should be maintained in ONGOING
<<goal>> The reviewers should be able to record that the three-months meeting with the student has happened and its result
<<goal>> The supervisors and the Master Thesis Committee should be able to update ONGOING
Ensure that students can choose a thesis correctly
<<goal>> Students should be able to choose an external thesis
<<goal>> Students should be able to choose an ad hoc thesis
<<goal>> Students should be able to choose a proposed thesis
<<goal>> Faculty members should be able to insert/remove a thesis in PROPOSALS
<<goal>> Students should be able to choose a thesis in PROPOSALS
Ensure that the final exam runs correctly
<<goal>> The supervisors should be able to submit the thesis document of a student enrolled in the final exam
<<goal>> The Final Exam Committee should be able to schedule the technical defense
<<goal>> Supervisors and reviewers should be able to provide the Final Exam Committee with the reports on the thesis of the students enrolled in the final exam
<<goal>> It should be possible to record the final exam dates and the composition of the corresponding Final exam committees
<<goal>> Ensure that the Final Exam Committee knows the weighted average of the marks for the students enrolled in the final exam
<<goal>> Ensure that the computed weighted averages and the reports are provided to the Final exam committee 10 days before the final exam
<<goal>> Ensure that the students enrolled in the final exam provide their exams' marks
<<goal>> Students working on an ongoing thesis should be able to enroll in the final exam
<<goal>> Students working on an ongoing thesis should be able to enroll in the final exam providing their exams' marks
Reduce the effort of the Master Thesis Committee in running the "Getting the degree process"
<<goal>> It should be possible to define the Master thesis Committee composition
<<goal>> Support the interaction among the Master Thesis Committee members
<<technological>> By means of a forum generating an email for each post
<<goal>> The Master thesis committee should be able to monitor the running of the "Getting the degree process"
<<goal>> A dashboard should be made available to the Master thesis committee to monitor the running of the "Getting the degree process"
Statistical information about the running of the "Getting the degree" process should be made available
<<goal>> The relevant info about running the "Getting the degree" process should be recorded
<<goal>> A report containing the descriptive statistics about the running of the "getting the degree" process should be produced when required by the Master thesis committee
<<privacy>> Students, faculty members, theses and final exam committees have different rights for accessing the info related to the "Getting the degree process"
<<goal>> Ensure that the insertion/removal of a thesis in PROPOSAL is be recorded plus when
<<goal>> Ensures that the refusals to revise a thesis should is record plus when
<<goal>> Ensure that the students' enrolments in final exams is be recorded
<<goal>> Discussed thesis should be recorded
<<goal>> The results of the final exams (weighted average, thesis marks and final marks) should be recorded
<<goal>> After the proclamation the results of the final exams (thesis marks and final marks) should be recorded
Powered By Visual Paradigm Community Edition

"correctly" means with respect to the "Getting the degree" process model

The majority of the leaf operative goal and part of the non functional goal has an associated specification

The mandatory activities are:
- reviewer confirming the three-month meeting between student and reviewer,
- supervisor uploading the thesis document 15 days before the final exam for a student enrolled in such exam,
- supervisor/reviewer providing the report on a thesis before the technical defense,
- student enrolling in the final exam one month before the end date of the thesis as recorded in ONGOING,
- Master thesis committee providing a reviewer two weeks after a thesis is inserted in ONGOING.

The dashboard should show:
- the complete info of each ongoing thesis,
- the current and past workload of each faculty member (number of supervised, revised and proposed theses, number of theses they is currently supervising and revising, and number of currently available theses proposed by them),
- the number of thesis proposals available for each topic

Who can see the following info relative to the "Getting the degree process":
- PROPASLS: anyone in the world
- ONGOING: students, faculty members and Master thesis committee
- thesis reports: the supervisors and the reviewers of the thesis, and the Final exam committee, and such reports must be destroyed after the final exam
- student's weighted average and thesis documents: the supervisors and the reviewers of the student's thesis, and the theses and final exam committees
- Dashboard on the current situation of the "Getting the degree process": Master thesis committee
- Statistics on the "Getting the degree process": Master thesis committee
- Discussed theses (title, student, supervisors, and final exam date): anyone in the world

The report should contain:
- the essential info (title, student to whom is assigned, start and end date, supervisors, and reviewers, topic, type) of each ongoing and each archived thesis,
- histograms showing the data about the time needed to complete a thesis, the thesis marks, and the student weighted average of the exams marks and their mutual relationships (e.g. thesis marks partitioned by thesis duration),
- the current and past workload of each faculty member (number of supervised, revised and proposed theses, and number of theses they is currently supervising and revising, and number of currently available theses proposed by them),
- cooperation level of each faculty member (e.g. unmotivated refusal of thesis revision, missed deadlines, unanswered messages),
- the number of proposed thesis (past and currently available) for each topic).

Please refer to the following table for the requirements as they would be referred throughout the document by their respective IDs.

Requirement ID	Description
R1	A dashboard should be made available to the Master thesis committee to monitor the running of the “Masterize your degree” process.
R2	A report containing the descriptive statistics about the running of the "Masterize your degree" process should be produced when required by the Master thesis committee. Only the Master thesis committee may access such statistics.
R3	After the proclamation the results of the final exams (thesis marks and final marks) should be recorded.
R4	Anyone may see PROPOSALS.
R5	Ensure that anyone is informed about approaching mandatory activities. Use email for informing people Avoid to annoy people by sending too many messages.
R6	Ensure that the computed weighted averages and the reports are provided to the Final exam committee 10 days before the final exam. Only the supervisors, the reviewers and the Final exam committee may see the averages and the reports.
R7	Ensure that the Master Thesis Committee is informed when some mandatory activity was not performed.
R8	Faculty members should be able to insert/remove a thesis in PROPOSALS.
R9	It should be possible to define the Master thesis Committee composition.
R10	It should be possible to record the final exam dates and the composition of the corresponding Final exam committees.
R11	No error in the inputs must be tolerated.
R12	Only the Master Thesis Committee, the students, the supervisors and the reviewers may see what is in ONGOING.
R13	Students should be able to choose a thesis in PROPOSALS.
R14	Students should be able to choose an ad hoc thesis.
R15	Students should be able to choose an external thesis.
R16	Students working on an ongoing thesis should be able to enrol in the final exam providing their exams' marks.
R17	Supervisors and reviewers should be able to provide the Final Exam Committee with the reports on the thesis of the students enrolled in the final exam. Only the supervisors, the reviewers and the Final exam committee may see these reports, and such reports must be destroyed after the final exam.

R18	The supervisors should be able to submit the thesis document of a student enrolled in the final exam.
R19	The authentication of anyone must be done using the UNIGE LDAP system.
R20	Support the interaction among the Master Thesis Committee Members By means of a forum generating an email for each post.
R21	The Final Exam Committee should be able to schedule the technical defence.
R22	The Master thesis committee should be able to select a reviewer for each started thesis.
R23	The reviewers should be able to record that the three-months meeting with the student has happened and its result.
R24	The supervisors and the Master Thesis Committee should be able to update ONGOING.
R25	The system should be developed using the "Java" technologies.
R26	The system should be non-available at most for one hour in the two weeks before a final exam session.
R27	The types of thesis, the way the final exam is organized, how the weighted average is computed and the thesis marks may change.

Reviewing Inputs

The main goal of this step is to review and identify the requirements that are to be considered as drivers. The inputs are summarized in the table below.

The following requirements are chosen based on their impact on the architecture and the core business of the system.

Category	Details	Description
Design Purpose	This is a greenfield system and the purpose of this system is to produce a sufficiently detailed design to support the construction of the system.	-
Primary Functional Requirements	R13, R14, R15, R16, R8, R24, R18, R17, R21, R10, R22, R23, R6, R3, R9	Because they all support the core functionality of the system.
Non-Functional Requirements	R19, R25, R26, R27	Because they impact the architecture of the system.
Other Functional Requirements	R1, R2, R20, R5	Additional Features.
Other Non-functional Requirements	R11, R2, R7, R12, R20, R5, R4, R6, R17.	Involves requirements related to Privacy, UI, interaction.

🟨 1st and 2nd Iteration 🟩 3rd Iteration

Iteration 1: An overall system goal is Established

This is the first iteration in the design of a greenfield system, so the iteration goal is to achieve the architecture of the system by choosing the drivers that may influence the general structure of the system.

Step 2: Establishing Iteration goal by selecting drivers:

For this step, the requirements classified as Primary Functional Requirements and Non-Functional Requirements are considered.

Step 3: One or More Elements of the System are chosen to refine:

Since the system is a greenfield effort, so in this case the element to refine is the entire Masterize your degree system, which is shown in the figure 1, refinement of the system is through decomposition.

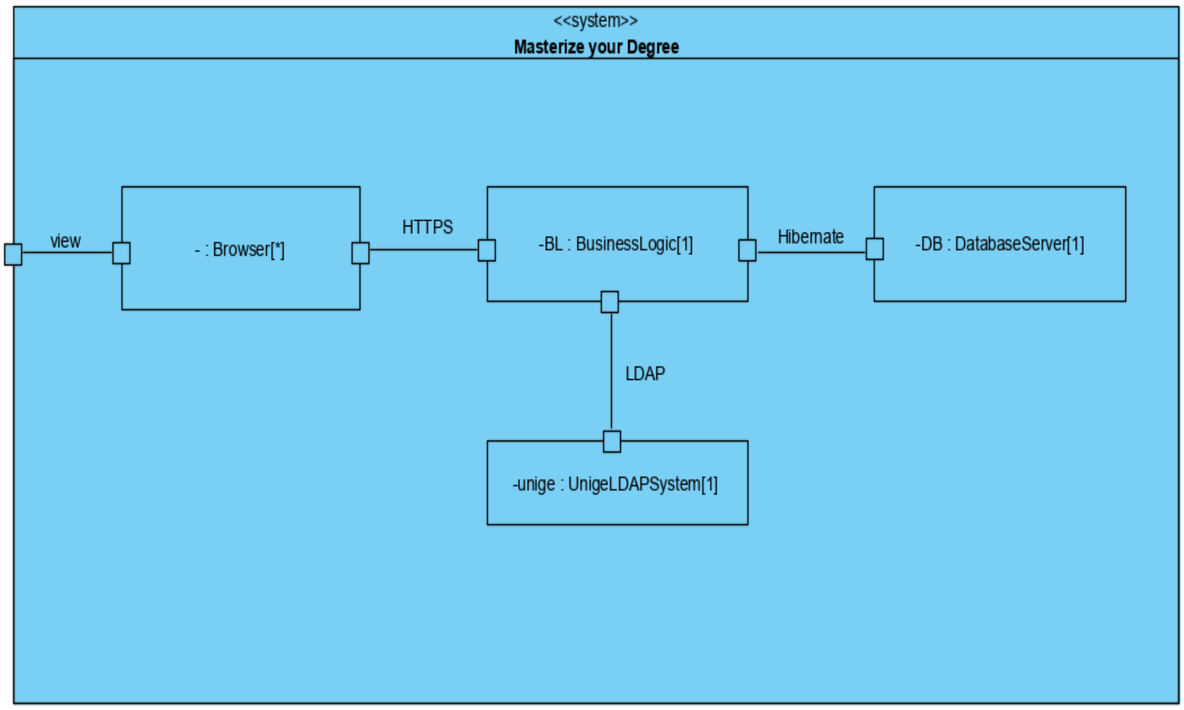


Figure 1: Composite Structure diagram for Masterize your degree system

Step 4: Design concepts that satisfy the drivers selected are chosen in this step

In this initial iteration, given the goal of structuring the entire system, design concepts are selected accordingly. The following table summarizes the selection of design decisions.

Design Decisions and Locations	Rationale
Logically structure the client part of the system using **Web Application** reference Architecture.	This reference architecture is oriented towards the development of applications that are accessed from a web browser and for applications that can adapt to whichever device the user is viewing on. Since this system does not need any fancy and rich interface, something as simple as a web application would do the job.

Discarded Alternatives:

Alternative	Reason for Discarding
Rich Internet applications	This reference architecture is oriented towards the development of applications with a rich user interface that runs inside a web browser. This option was discarded as "Masterize your degree" does not need rich/complex UI, and hence this option was discarded.
Mobile Applications	This reference architecture is oriented towards the development of applications that are deployed in handheld devices. This option was discarded considering the overhead of the installing the application on the devices.

Design Decisions and Locations	Rationale
Logically structure the server part of the system using the **Service Application** reference architecture.	Service applications do not provide a user interface but rather expose services that consumed by other applications. No other alternatives were considered and discarded, as the architect was familiar with this reference architecture and considered it fully adequate to meet the requirements.
Build the application using **Spring Java framework** and other UI technologies.	Spring framework is one of the standard frameworks for building service applications and web applications.

	Another reason for choosing Java based applications is due to the fact that the developers are well versed in it. R25, R27

Step 5: Instantiating Architectural Elements, Allocating Responsibilities, and Defining Interfaces

The next iteration has a better explanation about the elements and their functionalities since it is typically too early to define them here.

Step 6: Sketch Views and Record Design Decisions

The diagram in Figure 2 shows the sketch of a module view.

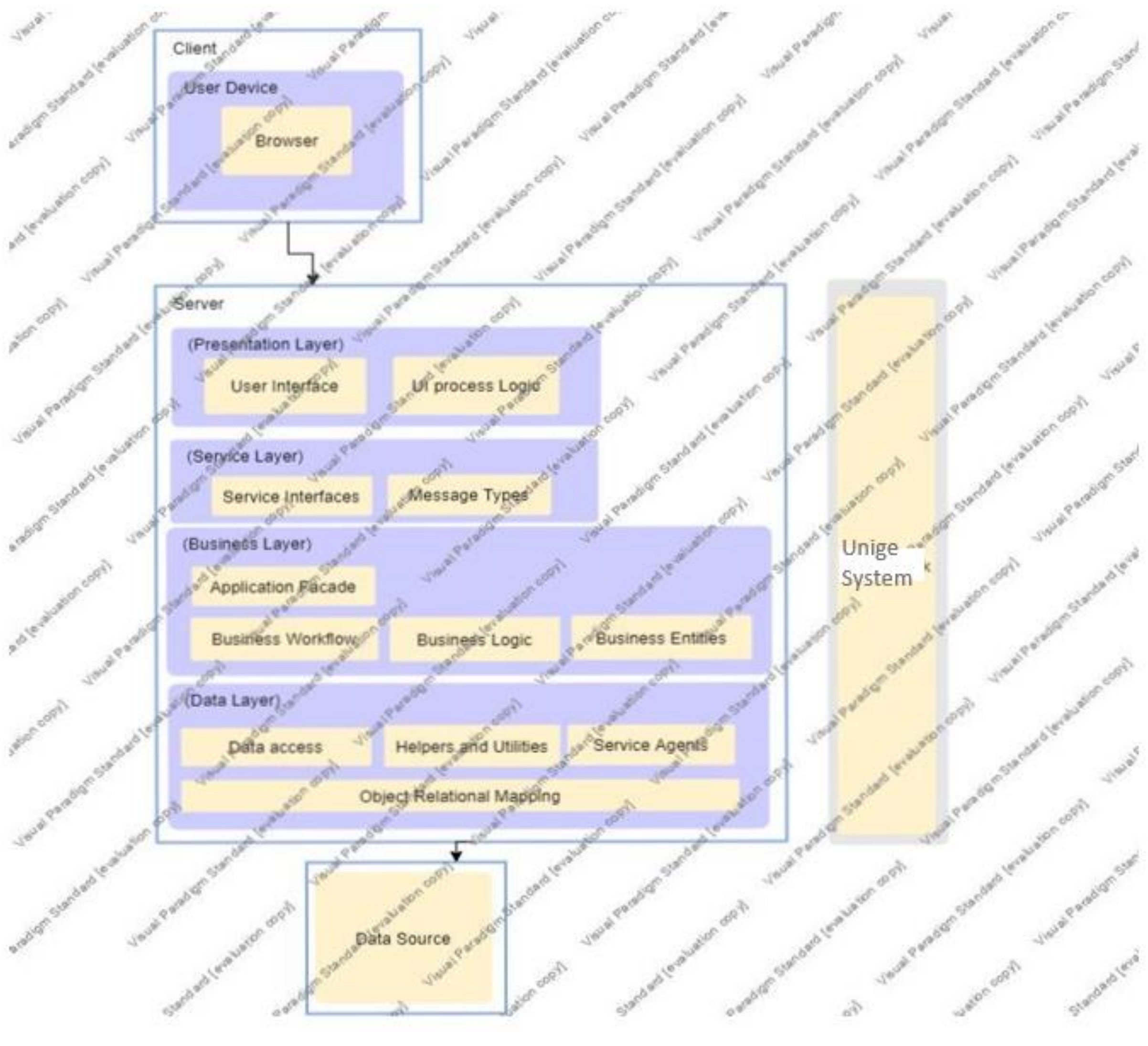


Figure 2: Module View of Masterize your Degree

The deployment diagram in Figure 3 sketches an allocation view that illustrates where components associated with modules in the previous diagram would be deployed.

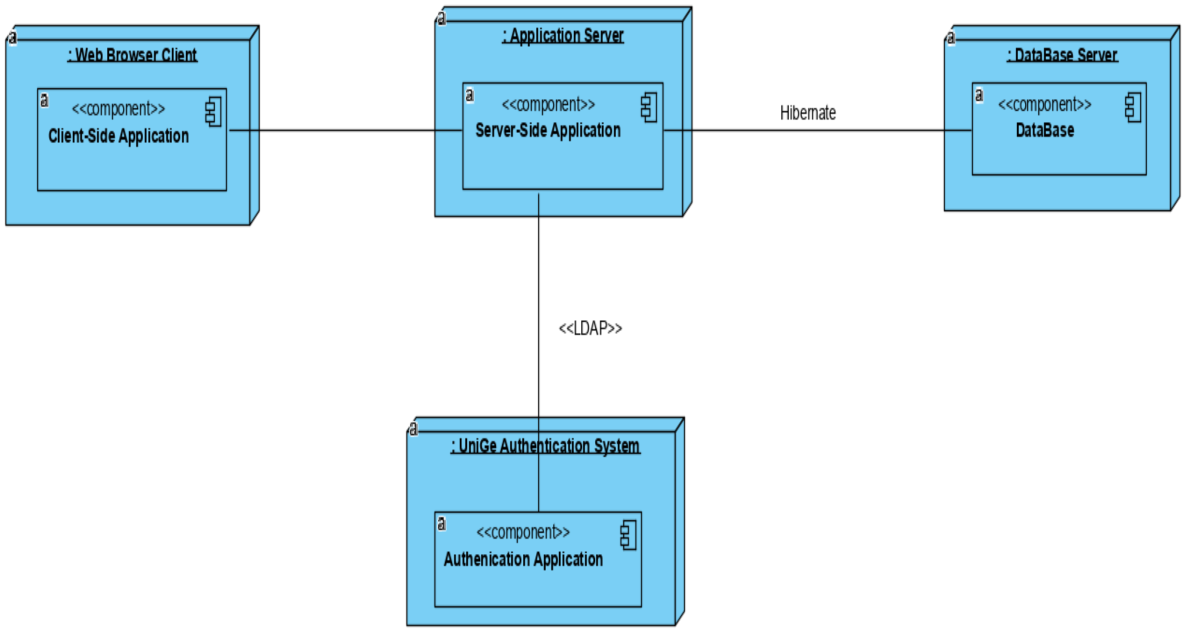


Figure 3: Initial deployment diagram of Masterize a degree system

The responsibilities of the elements are summarized here:

Element	Responsibility
Web Browser Client	A web browser, which hosts the client-side logic of the application.
Application Server	The server that hosts server-side logic of the application and also serves web pages.
Database Server	The server that hosts a relational database. Having a backup database would solve the risk of the database being destroyed up to an extent. R26
UniGe LDAP System	External system for authentication of users that would like to login to Masterize your degree system. R19

Also, information about relationships between some elements is summarized below

Element	Responsibility
Between web/app server and database server	Communication with the database will be done using JDBC protocol.
Between web/app server and UniGe System.	LDAP protocol is used for user authentication based on their profile inside the LDAP server. R19

Step 7: Performing Analysis of Current Design and Reviewing Iteration Goal and Achievement of Design Purpose

Please find the Kanban board here
https://trello.com/b/K8b43e0O/getting-a-degree.

Iteration 2: Identifying Structures to Support Primary Functionality

This section presents the results of the activities that are performed in each of the steps of ADD in the second iteration of the design process for Masterize your degree system. In this iteration, a more detailed description of the functionalities mentioned above in iteration 1 are focussed on, that will drive the implementation.

This movement from generic to specific is done because it's difficult to design everything upfront, and to show all the requirements in a more systematic way.

Step 2: Establish Goal by Selecting Drivers

The goal of this iteration is to address and identify structures to support primary functionality.

In this second iteration, the same requirements as the above iteration are considered to show a detailed description.

Step 3: Choosing One or More Elements of the System to Refine

The elements that will be refined in this iteration are the modules located in the different layers defined by the two reference architectures from the previous iteration. In general, the support of functionality in this system requires the collaboration of components associated with modules that are located on different layers.

Step 4: Choosing One or More Design Concepts that Satisfy the Selected Drivers

In this iteration, several design concepts are used to show the design decisions. The following table summarizes the design decisions.

Design Decisions and Location	Rationale and Assumptions
Entity Relationship Diagram	The entities that participate in the primary use cases are identified and modelled using E-R diagrams to accelerate the phase of design.
Mapping the system requirements to the class diagram of the application.	Moving from an abstract representation of the data model to a static structure and behaviour of the proposed system using class diagrams.
Connect components associated with modules using **Spring Boot**.	This framework uses an inversion of control approach that allows different aspects to be supported and the modules to the unit-tested.
Associate frameworks with a module in the data layer.	ORM Mapping is encapsulated in the modules that are contained in the data layer. This **Hibernate** framework previously selected is associated with these modules.

While the structures and interfaces are identified in this step of the method, they are captured in the next step.

Step 5: Instantiating Architectural Elements, Allocating Responsibilities, and Defining Interfaces

Design Decisions and Location	Rationale and Assumptions
Data Access Object Pattern (structural pattern) for accessing data.	The functionality of this pattern is to hide from the application all the complexities involved in performing CRUD operations in the underlying storage mechanism. This permits both layers to evolve separately without knowing anything about each other.
Data transfer object design pattern used to transfer data between software application subsystems.	Since each call to any remote interface is expensive, response to each call should bring as much data as possible. So, if multiple requests are required to bring data for a particular task, data to be brought can be combined in a DTO so that only one request can bring all the required data.
Strategy Pattern for defining the manner in which communication between classes or entities takes place.	You can isolate the implementation details of an algorithm from the code that uses it. (Loose coupling which is compatible with spring class relation concepts). Used for accessing functionalities with different objects. (DI) Discarded AI Alternative | Reason for discarding **Façade Pattern** | It's mainly used for accessing multiple objects by creating an interface above other objects.

Step 6: Sketching Views and Recording Design Decisions

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

Please refer to the masterizeyourdegree.vpp file for E-R diagram.

For class diagrams please refer to the file masterizeyourdegree.vpp.

Step 7: Performing Analysis of Current Design and Reviewing Iteration Goal and Achievement of Design Purpose

Please find the Kanban board here https://trello.com/b/K8b43e0O/getting-a-degree.

Iteration 3: Addressing the user interaction, Privacy requirements and Communication

This section presents requirements that are aggregated which have been addressed in the last two iterations but partially or requirements that exist in interaction or privacy category and have not been mentioned yet.

These requirements are categorized into three:

Privacy which defines the authentication and role management;

Interaction which involves the UI design;

Communication which involves the requirements that fall under email system or forums for interaction.

Step 2: Establish Iteration Goal by Selecting Drivers

For this iteration, privacy and role management requirements, interaction, and informing requirements are focused on.

Step 3: Choosing One or More Elements of the System to refine

For this step, the elements that will be refined are elements of the business logic layer and the presentation layer which provide additional functionalities to the already present system.

Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

The design concepts used in this iteration are the following:

Design Decision and Location	Rationale and Assumptions
For control access we use the **LDAP RBAC**.	In this way we can create, delete users from the system also for the permissions, defining new roles and access methods for the system. R6, R12, R4
For informing system, **Spring Mail** system is used.	For providing Mail service for the system because it is integrated with Spring. R7, R2, R5, R20, R1 **Discarded alternatives:** **Alternative** — **Reason for Discarding** Java Mail Service — Increase in development time for custom code.
For requirements which involve things to be shown on the UI, **HTML** and **JavaScript** is suggested.	All the developers seemed to be familiar with **HTML** and **JavaScript**, and hence this is suggested.
JUnits, Integration Testing must be implemented for all the requirements.	Different scenarios for JUnits testing must be done by the developers. Integration Testing should be done to test the application as a whole and to see how they would behave with other modules. This can ensure that there would be no errors in the inputs. R11.

Step 5: Instantiating Architectural Elements, Allocate Responsibilities, and Define Interfaces

The diagram below shows the class diagram for email system:

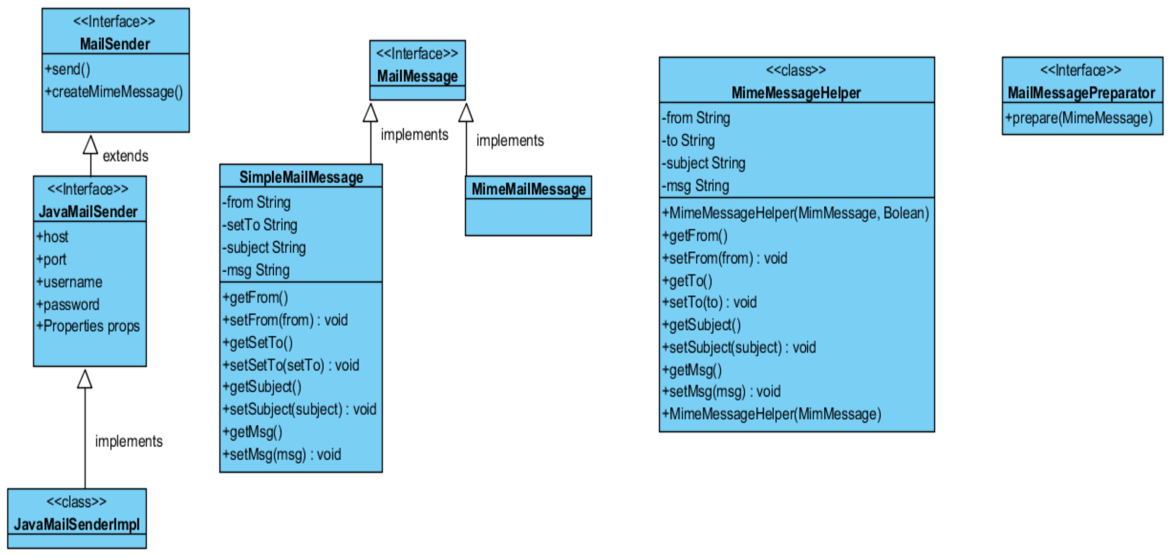


Figure 6: Class Diagram of Spring Mail system

Element	Responsibility
MailSender Interface	The top-level interface that provides basic functionality for sending simple emails
JavaMailSender Interface	It is the subinterface of MailSender. It supports MIME messages. It is mostly used with MimeMessageHelper class for the creation of JavaMail MimeMessage, with attachment etc. The spring framework recommends MimeMessagePreparator mechanism for using this interface.
JavaMailSenderImpl Class	It provides the implementation of JavaMailSender interface. It supports JavaMail MimeMessages and Spring SimpleMailMessages.
SimpleMailMessage Class	It is used to create a simple mail message including from, to, cc, subject and text messages.
MimeMessagePreparator Interface	It is the callback interface for the preparation of JavaMail MIME messages.
MimeMessageHelper Class	It is the helper class for creating a MIME message. It offers support for inline elements such as images, typical mail attachments and HTML text content.

MIME protocol– Multipurpose Internet Mail Extensions is a protocol to define the transferred content.

Step 6: Evaluation of the Requirements

Please find the Kanban board here https://trello.com/b/K8b43e0O/getting-a-degree.