**Software Requirements Specification**

**for**

**MSUClassIC**

**Version 1.0 approved**

**Prepared by Athul sajikumar,**

**Marcos lopez,**

**Samuel otlante**

**Team Shiners**

**Feb 24, 2023**

***Copyright © 2023 by Team Health. Permission is granted to use, modify, and distribute this document.***

***Software Requirements Specification for MSUClassIC***

***Page 2***

**Table of Contents**

**Table of Contents**

**Revision History**

1. **Introduction** 
   1. Purpose1
   2. Document conventions
   3. Intended audience and reading suggestions
   4. Product scope
   5. References
2. **Overall Description** 
   1. Product Perspective
   2. Product Functions
   3. Users Classes and Characteristics
   4. Operating Environment
   5. Design and Implementation constraints
   6. User Documentation
   7. Assumptions and Dependencies
3. **External Interface Requirements** 
   1. User Interfaces
   2. Hardware Interfaces
   3. Software Interfaces
   4. Communications Interfaces
4. **System Features** 
   1. User login
   2. View Schedules
   3. Request for class change
   4. Approve or deny requests
5. **Other Nonfunctional Requirements** 
   1. Performance Requirements
   2. Interface Requirements
   3. Maintainability Requirements
   4. Security Requirements
   5. Software Quality Attributes
   6. Business Rules
6. **Other Requirements**

**Appendix A: Glossary**

***Software Requirements Specification for MSUclassIC***

***Page 3***

**Appendix B: Analysis Models**

**Appendix C: To be Determined List**

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** | **Authors** |
| SRS Document | March 16,2024 | Initial Draft | 1 | Athul sajikumar,marcos lopez,Samuel otlande |
|  |  |  |  |  |

***Software Requirements Specification for MSUclassIC***

***Page 4***

**1.Introduction**

**1.1 Purpose**

*The purpose of this Software Requirement Specification document is to present a detailed description of the application known as ‘MSUClassIC’ (revision 1). It also describes precise and clear requirements and expectations for the application, providing a roadmap for developers to follow during the design and development process. The ‘MSUClassIC’ covered by this SRS is a web-based application that is designed majorly to help users schedule classes.*

**1.2 Document Conventions**

|  |  |
| --- | --- |
| convention | description |
| Font Type and size | Heading -Arial 18  Subheading -Arial 14  Body -Arial 11 |
| Application/System | MSUClassIC |
| UI | User interface |
| SRS | Software Requirement Specification |

**1.3 Intended Audience and Reading Suggestions**

*The intended audience for this SRS (Software Requirements Specification) includes developers, team lead, documentation writers(could be developers) and other stakeholders involved in the development and implementation of the software. The SRS document is organized into several sections, including an introduction, an overall description, requirements(functional and non-functional), and system features. The introduction and overall description section typically provide an overview of the project and its objectives, while the requirements section outlines the specific functional and non-functional requirements of the software. The system features will define all the features of the application in detail.*

*In terms of reading sequence suggestions, it is highly recommended to begin with the overview sections, including the introduction and purpose, to get an overall understanding of the project objectives and requirements. From there, developers may be interested in the technical details and specifications of the software, while the team lead may be more concerned with the scope timeline, and budget along with those of developers. Documentation writers may be interested in the functional and non-functional requirements that should be precisely included in user manuals and other documentation.*

***Software Requirements Specification for MSUclassIC***

***Page 5***

**1.4 Product Scope**

*MSUClassiC will be a scheduling app for MSUTexas. Our platform is made to simplify scheduling for Registrars, College Deans, and Department Chairs at Midwestern State University.*

*Users need to schedule every semester at MSUTexas, and this requires them to place a professor, class, and room at a specified time. This process is tedious and frustrating for many of the users currently.*

*Users shouldn’t have to deal with messy paperwork anymore. Our app will let them ask for room changes right there, avoiding manual updates and paperwork hassles. Whether it's the start of a new semester or just to make modifications to the existing schedules, our app has customizable options to fit every need.*

*With different views by building, department, and classrooms, it's simple to see what's available and make smart choices.*

**1.5 References**

*Below are the documents/web addresses that this document refers to at some place in this document –*

*User Interface style reference -*

<https://vuejs.org>

*Feasibility study -*

**2. Overall Description**

**2.1 Product Perspective**

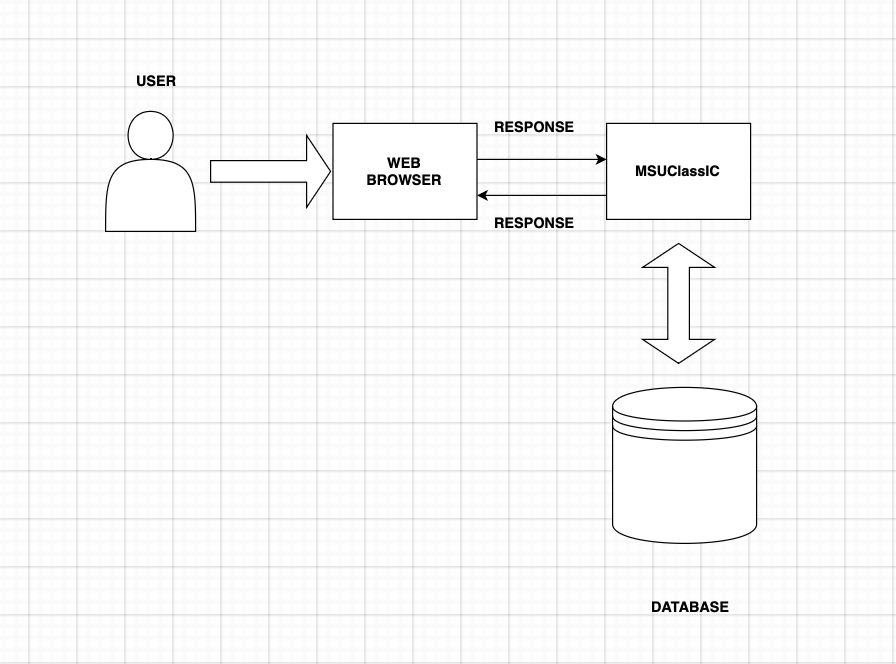
*The inspiration behind our class scheduling app stems from the growing importance of efficient academic management. Existing solutions in the market often prioritize complex functionalities and are monetized heavily. Our aim is to offer a user-centric approach, solely focused on enhancing the academic experience for users.*

***Software Requirements Specification for MSUclassIC***

***Page 6***

*Designed as a web application, our platform boasts a robust and intuitive interface, ensuring seamless navigation for all users. Positioned within the domain of "Academic Management," our app is dedicated to optimizing class scheduling processes for educational institutions.Our app empowers users with comprehensive tools to manage their academic schedules efficiently. By providing personalized views and streamlined workflows, we aim to simplify the scheduling experience, ultimately enhancing productivity and user satisfaction.*

*The below diagram shows a very high-level system diagram -*



**2.2 Product Functions**

*The significant functions the system must perform are listed below -*

*High priority requirements*

1. *The Chair will be able to log in and log out*
2. *The chair will be able to import excel file into web app*
3. *The chair will be able to view the excel contents in various pivot table views*
   1. *The chair will be able to view a pivot with professors vs time*
   2. *The chair will be able to view a pivot with professors vs courses*
   3. *The chair will be able to view a pivot with time vs courses*
   4. *The chair will be able to view a pivot with rooms vs courses*
   5. *The chair will be able to view a pivot with rooms vs time*
4. *The Chair will be able edit each view with a drag and drop motion*

***Software Requirements Specification for MSUclassIC***

***Page 7***

1. *The chair will be able submit changes as a request to the database*
2. *The chair will be able export database changes as an excel file*

**2.3 User Classes and Characteristics**

*Users are classified based on the type of product functions they use –*

**REGISTRAR**

1. *The registrar will be able to export the schedule to pdf or excels files*
2. *The registrar will be able to approve or deny changes to the schedule of class*
   1. *The registrar will be able to automatically approve any courses that do not have conflicts*
   2. *The registrar will be able to decide which courses are approves or denied when there are conflicts*
3. *The registrar will be able to view any conflicts of classes in a request queue*
4. *The registrar will be able to view any building by Room and Time*
5. *The registrar will be able to view any building by Course and Room*

**DEANS**

1. *All Deans will be able to approve or deny change to the schedule of classes for courses within their college*
   1. *All Deans will be able to automatically approve any course that do not have conflicts within their college*
   2. *All deans will be able to decide which courses are approves or denied when there are conflicts within their college*
2. *All deans will be able to view any conflicts of classes in a request queue within their college*
3. *All deans will be able to view any building by Room and Time for courses in their buildings*
4. *All deans will be able to view any building by Course and Room for courses in their buildings*

**CHAIR’S**

1. *The Chair will be able to log in and log out*
2. *The chair will be able to import excel file into web app*
3. *The chair will be able to view the excel contents in various pivot table views*
   1. *The chair will be able to view a pivot with professors vs time*
   2. *The chair will be able to view a pivot with professors vs courses*
   3. *The chair will be able to view a pivot with time vs courses*

***Software Requirements Specification for MSUclassIC***

***Page 8***

*4. The chair will be able to view a pivot with rooms vs courses*

*5. The chair will be able to view a pivot with rooms vs time*

1. *The Chair will be able edit each view with a drag and drop motion*
2. *The chair will be able submit changes as a request to the database*

***6.*** *The chair will be able export database changes as an excel file*

**2.4 Operating Environment**

*Operating environment for Calorie Tracker web application is listed below-*

*●  Operating system - Windows/MacOS*

*●  Database - MYSQL*

*●  Web Framework – VUE.JS*

*●  Language - Python*

*●  Frontend Technologies - Html,Css,Javascript*

*●  Software through which application is viewed - Web browser*

**2.5 Design and Implementation Constraints**

*Below are the design and implementation constraints -*

*●  The Language used to develop this software must be limited to either Python as per business request.*

*●  The Database to be used must be MYSQL ●  The product must be delivered by the end of the current semester as per business request*

*●  The database must return the queries in less than a second*

*●  The interface must be easy to use and navigate, with clear and concise instructions and*

*user-friendly layouts.*

*●  End user’s organization will be responsible for maintaining the delivered software* **2.6User Documentation**

***Software Requirements Specification for MSUclassIC***

***Page 9***

*The user documentation components (such as user manuals, online help, and tutorials) that will be delivered along with the software will include a project demo provided by the development team. This will include a YouTube link with audio that will demonstrate the functionality of the application, features, and capabilities the system would be capable of. A PowerPoint presentation will similarly display those features of the application.*

**2.7 Assumptions and Dependencies**

*Assumed factors that could affect the requirements stated in the SRS could include:*

1. *For a development environment, it is assumed that the developer has Python, Django, and Visual studio code installed and running on the machine*
2. *A browser that supports robust rendering of CSS, HTML & JavaScript must be used by the clients to see the product with the best appearance and performance.*
3. *It is assumed that the project will be completed within the agreed budget and time*
4. *It is assumed that the code will be developed using the best coding standards possible*
5. *It is assumed that a user is a person who understands the English language and knows basic usage of a web browser*

**3. External Interface Requirements**

**3.1 User Interfaces**

*The logical characteristics of each interface between the software product and the users can be described in terms of the following elements:*

* 1. *Input Requests from Deans and Chairs: The input elements for our course scheduling application encompass the information that Deans and Chairs need to provide to initiate scheduling requests. These elements include buttons, selection boxes, and other user interface controls. It's imperative that these input elements are designed to be clear, user-friendly, and indicative of the required information. Deans and Chairs should find it intuitive to input scheduling requests, ensuring efficient communication of their requirements.*
  2. *Output for Registrar's Approval: The output component of our application pertains to the information provided by the system in response to scheduling requests submitted by Deans and Chairs. This output includes text-based notifications and possibly graphical representations, depending on the specific context. The output must be clear, accurate, and appropriate for the scheduling task at hand. Ultimately, it should facilitate Registrar's decision-making by presenting scheduling requests in a comprehensible format.*
  3. *Screen Layout Constraint: All of the pages must follow the same header and footer and consistent typography has to be followed throughout the UI pages.*

***Software Requirements Specification for MSUclassIC***

***Page 10***

**3.2 Hardware Interfaces**

*Below are the logical and physical characteristics of each interface between the software product and the hardware components of the system -*

*●  Supported device types - Windows 10 or MacOS computer.*

*●  Communication protocols: The system uses HTTP communication protocols over the internet to send and receive data to the client aka the web browser.*

*● A browser that supports robust rendering of CSS, HTML & JavaScript*

**3.3 Software Interfaces**

*Below are the interface that is anticipated to be dealt with during this project -*

***User interface (UI)****, A web-based interface that allows users to interact with the system*

***Database interface.*** *This is the interface between the database and software which allows the storage and retrieval of data*

***Third-party Datasets,*** *MSUClassIC may use* ***Datasets*** *from external sources such as course schedule databases to enhance its functionality.*

**3.4 Communications Interfaces**

*Below are the requirements that are required for this product -*

***The web browser*** *MSUClassIC should be compatible with popular web browsers such as Firefox, chrome and brave .This provides an interface for users to navigate and interact with the application*

***Network Interface:*** *This allows the software to communicate with other devices over the internet and it should support protocols such as TCP/IP, HTTP, HTTPS*

***Security Requirement****. The system should require members to authenticate using secure strong passwords*

***Software Requirements Specification for MSUclassIC***

***Page 11***

**4. System Features**

*Below are the functional requirements for the product by system features, the major services provided by the product –*

**4.1 user login**

**Priority 9**

* + 1. *Description and Priority  
       All users will be able to log into MSUClassIC*
    2. *Stimulus/Response Sequences for Login:*
* *The user navigates to the login page of the application*
* *The user enters the credentials in to the designated fields*
* *The user clicks on the login button*
* *The verifies the entered credentials against the stored user database*
* *If the credentials are valid the system grand access to the user account*
* *If invalid the system displays an error message*
  + 1. *Functional Requirements*

*The system must allow the user to sign-up*

* 1. **View Schedules**

**Priority 9**

* + 1. *Description and Priority  
       The users will be able to view there own personalized views*
    2. *Stimulus/Response Sequences for viewing schedule:*
* *The user accesses the course scheduling application.*
* *The user navigates to the "View Schedules" section of the application.*

***Software Requirements Specification for MSUclassIC***

***Page 12***

* *The user selects a specific timeframe or filters the schedule based on certain criteria.*
* *The user clicks on a specific schedule entry to view detailed information. The user navigates through different pages or sections within the schedule viewing interface.*
* *The user performs actions such as printing or exporting schedules.*

4.3.3 *Functional Requirements*

*The system must allow the user to view schedules*

* 1. **Request for a class change**

**Priority 9**

* + 1. *Description and Priority*

*The user’s will be able to request for a class change*

* + 1. *Stimulus/Response Sequences for requesting class change:*
* *The user accesses the course scheduling application.*
* *The user specifies the desired changes, such as altering the class time, location, or instructor.*
* *The user submits the class change request by clicking on the "Submit" or equivalent button.*
* *The user receives a confirmation email or notification regarding the submission of the class change request.*
  + 1. *Functional Requirements*

*The system must allow the user to request for a class change*

* 1. **Approve or deny request**

**Priority 9**

* + 1. *Description and Priority*

*The user’s will be able to approve or deny request as per the user role*

***Software Requirements Specification for MSUclassIC***

***Page 13***

*4.4.2 Stimulus/Response Sequences for approving/denying request:*

* The registrar accesses the course scheduling application and logs into their account.
* The registrar navigates to the "Requests" or "Pending Approvals" section of the application.
* The registrar selects a specific request from the list to review.
* The registrar reviews the request and evaluates its merits based on scheduling conflicts, resource availability, and other relevant factors.
* If approving the request, the registrar clicks on the "Approve" or equivalent button.
* If denying the request, the registrar clicks on the "Deny" or equivalent button.
  + 1. *Functional Requirements*

*The system must allow the user to approve or deny request*

**5.Other Nonfunctional Requirements**

*●****Maintainability*** *- The interface should be easy to maintain and update, with clear and well-documented code and design.*

*●****Compatibility*** *-The app must be compatible with multiple devices and browsers*

*●****Scalability-*** *It should be able to support multiple users without performance degradation*

*●The* ***interface*** *must be easy to use and navigate, with clear and concise instructions and user-friendly layouts. The interface must not be crowded.*

**5.1 Performance requirements**

*The system must be able to handle multiple users without performance decrease*

**5.2 Interface requirements**

* *Must be a web based application*
* *The interface must be easy to use and navigate with clear and concise instructions and user friendly layouts*

***Software Requirements Specification for MSUclassIC***

***Page 14***

**5.3 Maintainability requirements**

*The interface should be easy to maintain and update*

**5.4 Security Requirements**

*The system should have appropriate measures in place to protect user data and to prevent unauthorized access*

**5.5 Software Quality Attributes**

*Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some of them are –*

* *MSUClassIC will be easy to use and navigate, it will have a simple and intuitive user interface that allows users to track their calorie intake with minimal effort . The interface will not be crowded.*
* *Compatibility: MSUClassIC is a web application that will be compatible with most web browsers running on a variety of devices and platforms, including laptops,desktops.*

**5.6 Business Rules**

*Our course scheduling application offers seamless access to scheduling information, allowing users such as Deans, Chairs, and Registrars to effortlessly view schedules without the need for login credentials. For enhanced functionality, users have the option to register and create accounts within the system, providing personalized access to scheduling features. Role-based access control ensures that each user is assigned specific permissions tailored to their responsibilities, with administrators, like Registrars, endowed with elevated privileges to manage scheduling requests and oversee overall operations. Through the application, users can submit scheduling requests, which are meticulously reviewed and managed by Registrars to ensure alignment with scheduling constraints and considerations. Timely notifications keep users informed of approval statuses and any updates to their scheduling requests, fostering effective communication and collaboration within the scheduling process. Robust security measures safeguard scheduling data and user information, ensuring data integrity and confidentiality in accordance with privacy regulations. With these features in place, our application streamlines scheduling operations, facilitating efficient management and coordination across all user roles.*

***Software Requirements Specification for MSUclassIC***

***Page 16***

1. **Other Requirements**

***Support Requirements***

*MSUClassIC will be supported by the development team for at least 3 months after its initial release around end of April 2024. Documentation training; The user manual will be available on*

**Appendix A: Glossary**

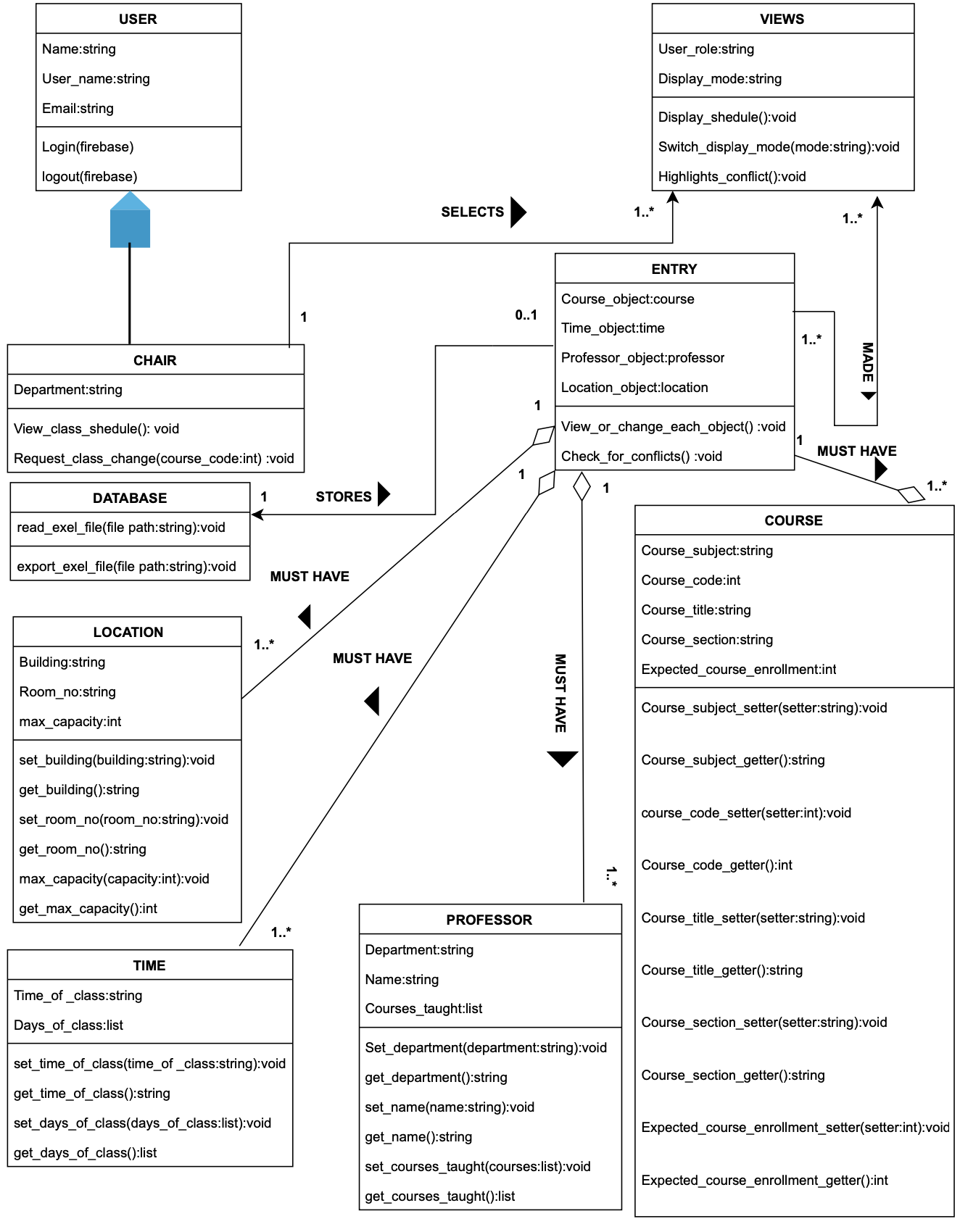
*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

|  |  |
| --- | --- |
| *Convention* | *Description* |
| *Font Type & Size (throughout this document)* | *Heading - Arial 18 Subheading - Arial 14 Body - Arial 11* |
| *UI* | *User Interface* |
| *Database* | *This is a collection of processed information that is essential to the system users.* |
| *Dataset* | *This a collection of organized data that is presented in a structured format is stored in various forms such as CSV,JSON,XML* |
| *SRS* | *Software Requirements Specification* |
| *Framework* | *A collection of libraries, tools, and modules that developers can use to speed up the software development process.* |
| *User* | *Any person who has access the web application, they can be registered users or non registered* |
| *UML class diagram* | *This structure diagram describes the structure of this system showing the system's classes, their attributes, operations , and the relationships among objects.* |
|  |  |

***Software Requirements Specification for MSUclassIC***

***Page 16***

**Appendix B: Analysis**

****

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Document \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*