

**FPT enrollment**

**management**

**Report 3 – Software Requirement Specification**

– Hanoi, August 2023 –

**Table of Contents**

[I. Record of Changes 3](#_Toc171542290)

[II. Software Requirement Specification 4](#_Toc171542291)

[1. Requirement Overview 4](#_Toc171542292)

[1.1 Context Diagram 4](#_Toc171542293)

[1.2 User Requirements 5](#_Toc171542294)

[1.3 System Functionalities 9](#_Toc171542295)

[2. Non-Functional Requirements 13](#_Toc171542296)

[3. Key Requirements for CRM System for Managing Student and Parent Data 16](#_Toc171542297)

[6. Design Principles Followed in the CRM System 16](#_Toc171542298)

# I. Record of Changes

|  |  |  |
| --- | --- | --- |
| Date | A\* M, D | In charge |
| 6/7/2024 | A | 1:Requirement Overview |
| 6/7/2024 | A | a:List use-case |
| 6/7/2024 | A | 1.1 Context Diagram |
| 6/7/2024 | A | 1.2 User Requirements |
| 6/7/2024 | A | 1.2.1 Actors |
| 6/7/2024 | A | 1.2.2 Use Cases |
| 6/7/2024 | A | 1.3 System Functionalities |
| 6/7/2024 | A | 1.3.1 Screens Flow |
| 6/7/2024 | A | 1.3.2 Screen Descriptions |
| 6/7/2024 | A | 1.3.3 Screen Authorization |
| 6/7/2024 | A | 1.3.4 Non-UI Functions |
| 6/7/2024 | A | 1.3.5 Entity Relationship Diagram |
| 6/7/2024 | A | 2. Non-Functional Requirements |
| 6/7/2024 | A | 1.1 External Interfaces |
| 6/7/2024 | A | 1.2 Quality Attributes |
| 6/7/2024 | A | 2.1: Functional Requirements |
| 6/7/2024 | A | 3. Key Requirements for CRM System for Managing Student and Parent Data |
| 6/7/2024 | A | 6. Design Principles Followed in the CRM System |

\*A - Added M - Modified D - Deleted

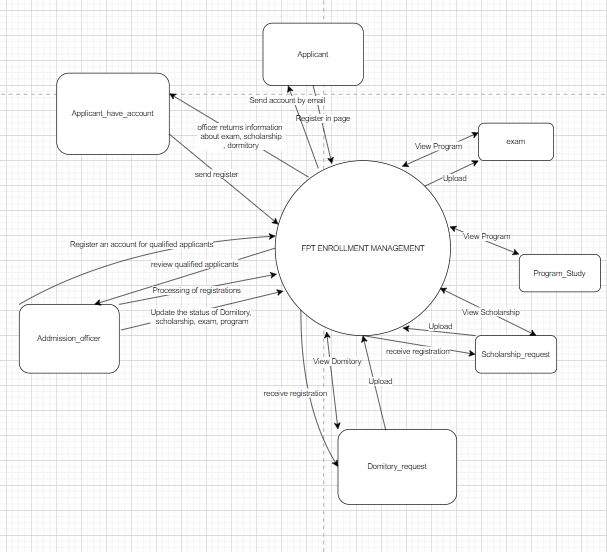
# II. Software Requirement Specification

## Requirement Overview

a:List use-case

|  |  |
| --- | --- |
| **Primary Actor** | **Description** |
| Applicant | Applicants pass through the admissions website to admission\_officer for approval |
| Applicant\_have\_account | The person whose application has been accepted by admission\_officer |
| Admission\_officer | Be the admin as well as the person who approves applicants' applications and account holders' registration applications |

### 1.1 Context Diagram



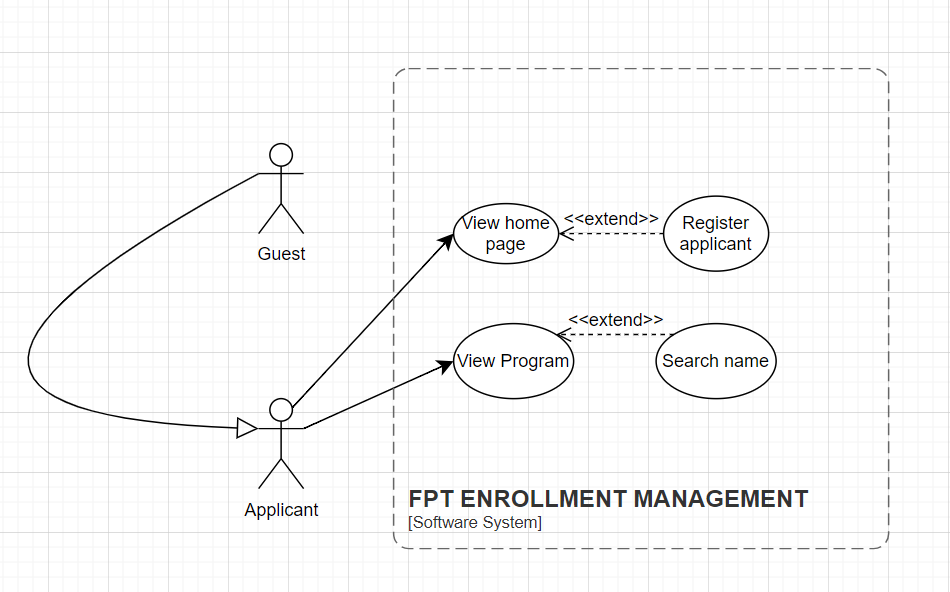
>>

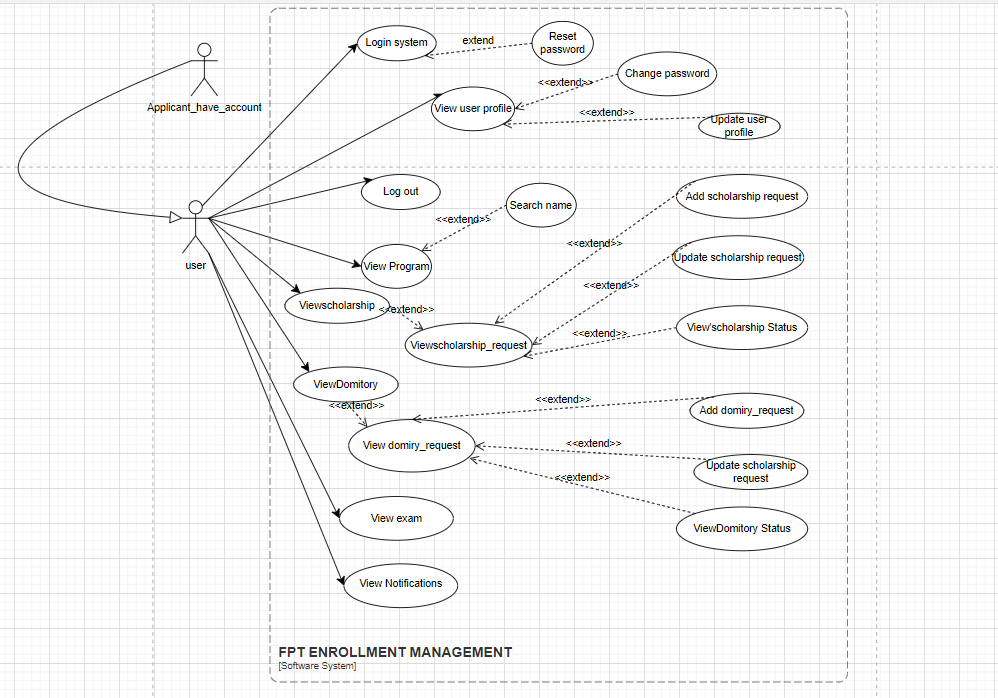
### 1.2 User Requirements

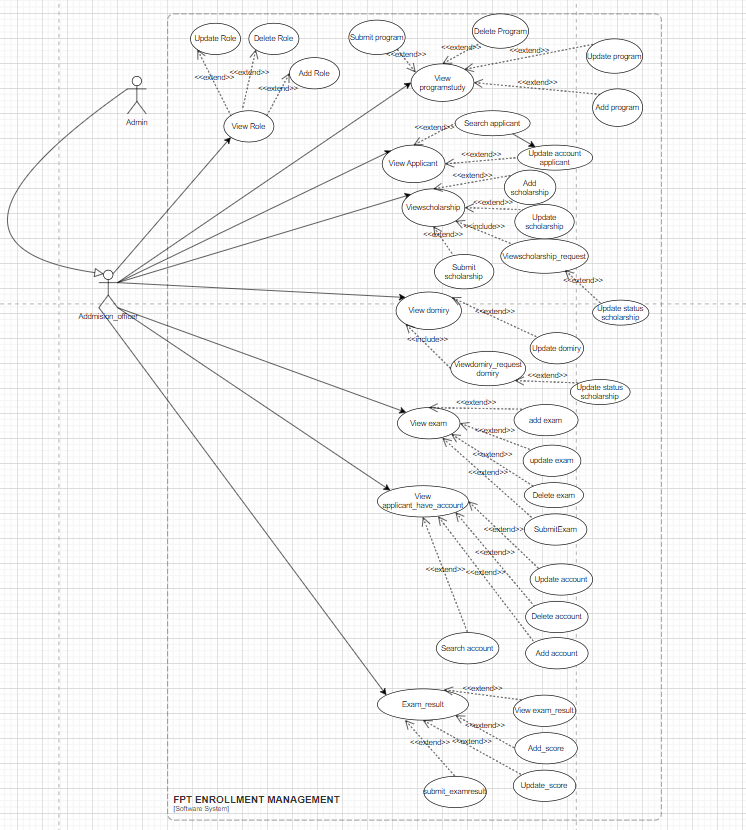
#### 1.2.1 Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Use Case** |
| 1 | Applicant | -Register applicant  -View Program Study |
| 2 | Applicant\_have\_account | -login sytem  -resetpassword  -View user profile  -change password  -update user profile  -log out  -view program  -search program  -viewscholarship  -viewscholarship\_request  -add scholarship request  -update scholarship request  -view scholarship status  -view domitory  -view domitory\_request  -add domitory\_request  -update domitory\_request  -view domitory\_status  -view exam  -View notifications |
| 3 | Admission\_officer | -update role  -delete role  -add role  -submit program  -delete program  -update program  -add program  -view applicant  -search applicant  -update account applicant  -add scholarship  -update scholarship  -submit scholarship  -update status scholarship  -update domitory  -update status domitory  -add exam  -update exam  -delete exam  -view applicant\_have\_account  -update account  -delete account  -add account  -search account  -add\_score  -update\_score  -submit\_examresult |

#### 1.2.2 Use Cases

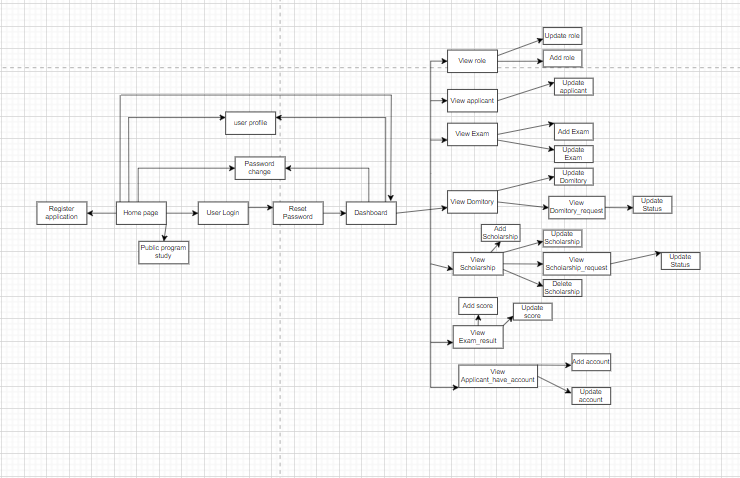
**

**



### 1.3 System Functionalities

#### 1.3.1 Screens Flow



#### 1.3.2 Screen Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Register applicant | Home page | Applicants can fill in their details to apply without registering an account. |
| 2 | Program study | Home page | Allows viewers to see the school's majors |
| 3 | Prassword change | User profile | Change user password |
| 4 | Reset password | Login | Change the password on the login page when the user forgets by authenticating via email |
| 5 | Dashboard |  | The main hub of the system after logging in, providing access to various functionalities. |
| 6 | User Profile | Homepage | Allows users to view and update their profile information. |
| 7 | Update user Profile | User profile | Allows users to update their personal information. |
| 8 | View Notifications | User profile | Change user password |
| 9 | View Applicant | View Applicant | Allows admins to view the list of applicant. |
| 10 | View role | View role | Allows admins to view the list role |
| 11 | View exam | View exam | Allows user to view exam |
| 12 | View Domitory | View Domitory | Allows user to view domitory |
| 13 | View Scholarship | ViewScholarship | Allows user to view Scholarship |
| 14 | View Exam\_result | View Exam\_result | Allow user to view Exam\_result |
| 15 | View applicant\_have\_account | View applicant\_have\_account | Allow admin to view applicant\_have\_account |
| 16 | Add role | View role | Add information role by admin |
| 17 | Update role | View role | Update information role |
| 18 | Update applicant | View applicant | Edit qualified applicants to account holders |
| 19 | Add Exam | View Exam | Add information exam by admin |
| 20 | Update Exam | View Exam | Update information exam by admin |
| 21 | Update Domitory | View Domitory | Update information Domitory by admin |
| 22 | View Domitory\_request | View Domitory\_request | Allow user to view your request,admin view all |
| 23 | Update Status domitory | View Domitory\_request | Update status request by admin |
| 24 | Add scholarship | View Scholarship | Add scholarship by admin |
| 25 | Update scholarship | View Scholarship | Update scholarship by admin |
| 26 | View scholarship\_request | View Scholarship\_request | Allow user to view your request,admin view all |
| 27 | Update status scholarship | View Scholarship\_request | Update status by admin |
| 28 | DeleteScholarship | View Scholarship\_request | Delete Scholarship by admin |
| 29 | Add score | View Exam\_result | Add Score by the admin |
| 30 | Update score | View Exam\_result | Update Score by the admin |
| 31 | Add account | View Applicant\_have\_account | Add account applicant by the admin |
| 32 | Update account | View Applicant\_have\_account | Update information account by admin |

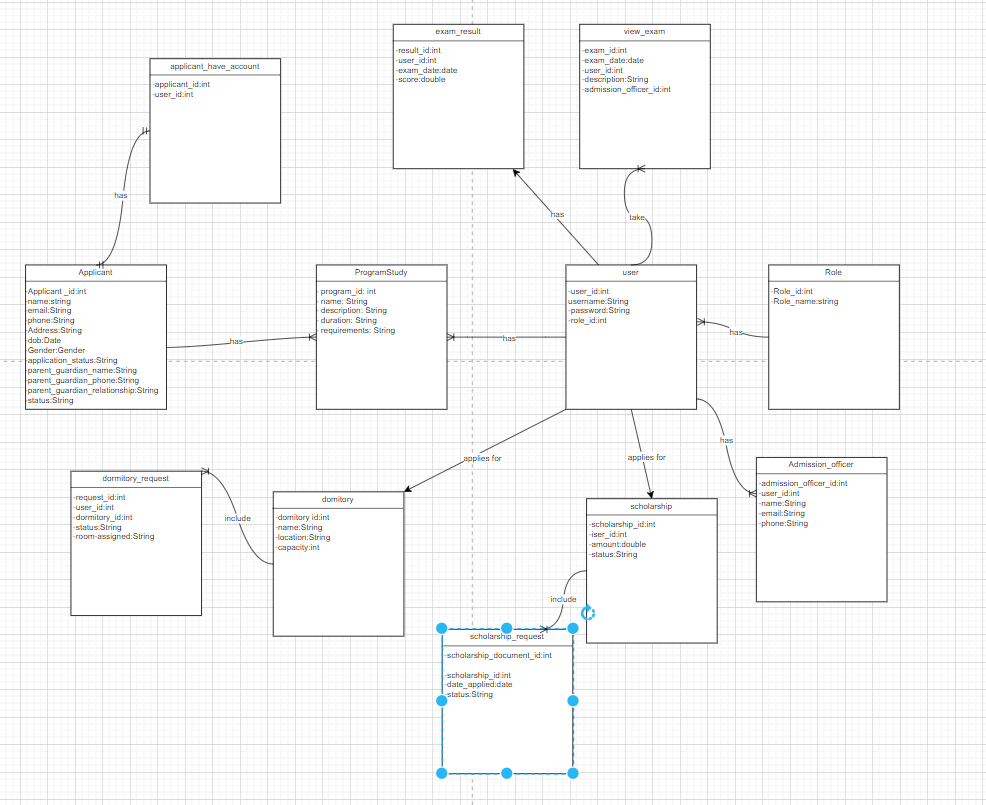
#### 1.3.3 Screen Authorization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Applicant\_have**  **\_account** | **Admin** | **Applicant** | **…** |
| Register applicant | X | X | X | ... |
| Program study | X | X | X | ... |
| Password change | X | X |  | ... |
| Reset password | X | X |  | ... |
| Dashboard | X | x |  | ... |
| User Profile | X | X |  | ... |
| Update User Profile | X | X |  | ... |
| View Notifications | X | X |  | ... |
| View Applicant | X |  |  | ... |
| View Role | X |  |  | ... |
| View Exam | X | X |  | ... |
| View Dormitory | X | X |  | ... |
| View Scholarship | X | X |  | ... |
| View Exam Result | X | X |  | ... |
| View Applicant\_have\_account | X |  |  | ... |
| Add Role | X |  |  | ... |
| Update Role | X |  |  | ... |
| Update Applicant | X |  |  | ... |
| Add Exam | X |  |  | ... |
| Update Exam | X |  |  | ... |
| Update Dormitory | X |  |  | ... |
| View Dormitory\_request | X |  |  | ... |
| Update Status Dormitory\_request | X |  |  | ... |
| Add Scholarship | X |  |  | ... |
| Update Scholarship | X |  |  | ... |
| View Scholarship\_request | X |  |  | ... |
| Update Status Scholarship\_request | X |  |  | ... |
| Delete Scholarship | X |  |  | ... |
| Add Score | X |  |  | ... |
| Update Score | X |  |  | ... |
| Add Account | X |  |  | ... |
| Update Account | X |  |  | ... |

#### 1.3.4 Non-UI Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| **1** | **Batch/Cron Job** | **Backup Dữ liệu** | **Thực hiện sao lưu dữ liệu hệ thống tự động theo chu kỳ định sẵn.** |
| **2** | **Service** | **Gửi Thông báo qua Email** | **Gửi thông báo qua email đến người dùng về các sự kiện và hành động trong hệ thống.** |
| **3** | **API** | **Quản lý Người dùng API** | **Cung cấp các endpoint để thực hiện các thao tác CRUD trên tài khoản người dùng, vai trò và quyền hạn.** |
| **4** | **Batch/Cron Job** | **Nhập Dữ liệu** | **Thực hiện định kỳ nhập dữ liệu từ bên ngoài vào cơ sở dữ liệu hệ thống.** |
| **5** | **Service** | **Dịch vụ Xác thực** | **Xử lý quá trình đăng nhập/đăng xuất và xác thực người dùng một cách an toàn.** |
| **6** | **API** | **API Báo cáo** | **Cung cấp các endpoint để tạo và truy cập các báo cáo trong hệ thống.** |
| **7** | **Batch/Cron Job** | **Tác vụ Dọn dẹp** | **Chạy các nhiệm vụ định kỳ để dọn dẹp dữ liệu không sử dụng hoặc thực hiện các hoạt động bảo trì.** |

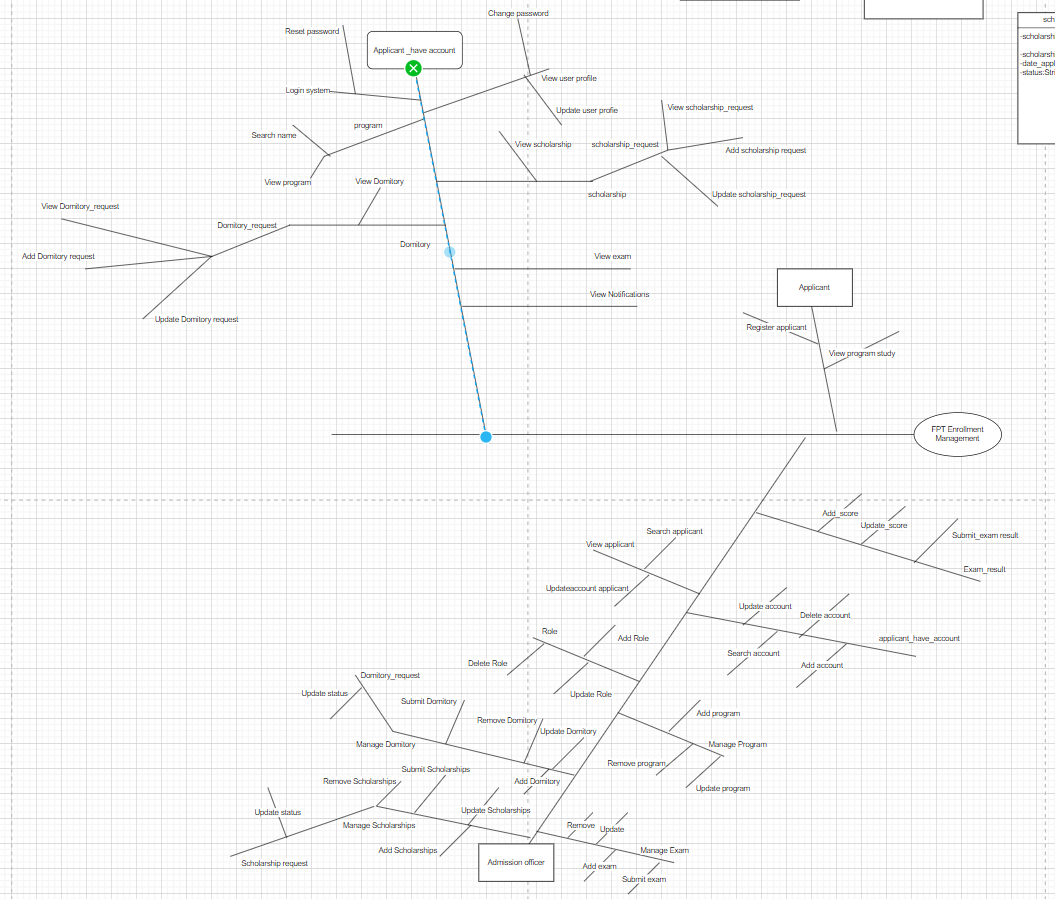
#### 1.3.5 Entity Relationship Diagram



**Entities Description**

|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | Applicant | Information of all study applicants |
| 2 | Applicant\_have\_account | The account of the person has been approved by the admin |
| 3 | Program\_study | Program information show all user |
| 4 | User | Contains the applicant have account and admission officer user accounts |
| 5 | Role | User account roles |
| 6 | Admission\_officer | Contains admin account information |
| 7 | Scholarship | Contains information of scholarships |
| 8 | Scholarship\_request | Contains user information when registering for a scholarship |
| 9 | Domitory | Contains information of the dormitory |
| 10 | Domitory\_request | Contains user information when registering a domitory |
| 11 | View\_Exam | Contains exam schedule information for applicants who already have an account |
| 12 | Exam\_result | Contains information about the date with points and the date with points for the user |

1.4:feature tree



## 2. Non-Functional Requirements

**1.1 External Interfaces**

This section provides information to ensure that the system will communicate properly with users and with external hardware or software/system elements.

**User Interfaces:**

1. **Responsive Web Interface:**
   * Accessible via major browsers (Chrome, Firefox, Safari, Edge).
   * Intuitive design with role-based access for applicants, admission officers, and users with accounts.
   * Multi-language support for diverse user preferences.

**Hardware Interfaces:**

2. **Server Infrastructure:**

* Hosted on scalable cloud platforms (e.g., AWS, Azure).
* Load balancing to manage high traffic efficiently.
* Secure communication protocols (HTTPS) to ensure data integrity and security.

**Software Interfaces:**

3. **Database:**

* Relational database (e.g.,SQL server) with support for sharding and replication.
* Secure and efficient ORM (Object-Relational Mapping) interaction with the database.

1. **APIs:**
   * RESTful APIs for external system interactions such as application submission, status updates, and retrieving program information.
   * Secure authentication mechanisms (e.g., OAuth 2.0).
   * Industry-standard data formats (e.g., JSON, XML).

**External Systems:** 5. **Email Services:**

* Integration with external email services (e.g., SendGrid, Mailgun) for notifications and confirmations.

1. **Authentication Services:**
   * Integration with external authentication services (e.g., Google OAuth, Facebook Login) for single sign-on capabilities.

**1.2 Quality Attributes**

This section lists all the required system characteristics (quality attributes) specification.

**1.2.1 Usability**

This section includes all those requirements that affect usability.

1. **Training Time:**
   * Normal users should become productive within 2 hours of training.
   * Power users should become productive within 1 hour of training.
2. **Task Performance:**
   * Measurable task times for typical tasks should be within industry standards.
   * For instance, an application submission should not take more than 5 minutes on average.
3. **Usability Standards:**
   * The system should conform to common usability standards such as IBM’s CUA standards and Microsoft’s GUI standards.
   * User satisfaction should be maintained at a minimum of 85% in usability surveys.

**1.2.2 Performance**

The system’s performance characteristics are outlined in this section.

1. **Response Time:**
   * Average response time for a transaction should be less than 2 seconds.
   * Maximum response time for any transaction should not exceed 5 seconds.
2. **Throughput:**
   * The system should handle at least 100 transactions per second under peak load conditions.
3. **Capacity:**
   * The system should support up to 10,000 concurrent users.
   * The database should be able to manage up to 1 million records without performance degradation.
4. **Resource Utilization:**
   * Memory usage should not exceed 70% of available RAM under normal operating conditions.
   * Disk utilization should not exceed 80% of available storage.
   * Network bandwidth utilization should be optimized to ensure minimal latency.

#### 2.1: Functional Requirements

Functional requirements describe the behavior of the system, including specific features and functionalities.

1. **User Registration and Authentication:**
   * Applicants can submit applications without registering an account.
   * Admission officers can create and manage applicant accounts, converting applicants to users with accounts (applicant\_have\_account).
   * Users with accounts can log in using their credentials.
   * Password reset functionality via email verification.
2. **Applicant Management:**
   * Applicants can fill out and submit application forms.
   * Applicants can view and update their application status.
   * Admission officers can view and manage applications.
   * Applicants can request scholarships and dormitories.
3. **Program Information:**
   * Users can browse available programs and their details.
   * Users can view admission requirements and program descriptions.
4. **Exam Management:**
   * Users can view scheduled exams and exam details.
   * Admin can add, update, and delete exam information.
   * Applicants can view their exam results.
5. **Role and User Management:**
   * Admin can create, update, and delete user roles.
   * Admin can assign roles to users.
   * Users can update their profile information.
6. **Notification System:**
   * Users receive notifications about application status changes.
   * Admin can send bulk notifications to users.
7. **Dormitory Management:**
   * Users can request dormitory accommodation.
   * Admin can view and update dormitory requests and status.
8. **Scholarship Management:**
   * Users can apply for scholarships.
   * Admin can view and manage scholarship requests and status.
9. **Dashboard:**
   * Provides a centralized hub for accessing various system functionalities.
   * Displays key information and metrics relevant to the user's role.

### 3. Key Requirements for CRM System for Managing Student and Parent Data

#### Functional Requirements

1. **Data Entry:**
   * **Applicant Registration:** Applicants can submit their details (name, email, phone, address, etc.) through the web portal.
   * **Admin Data Entry:** Admission officers can enter and update data for applicants, programs, exams, dormitories, scholarships, and more.
2. **Communication Management:**
   * **Record Communication:** Ability to log phone calls and emails between admission officers and students/parents.
   * **Automated Emails:** System can send automated emails for application status updates, exam schedules, and other notifications.
   * **Email Templates:** Predefined templates for common communications.
3. **Search and Update:**
   * **Search Functionality:** Users can search for applicants, programs, exams, dormitories, and scholarships based on various criteria (name, email, program, status, etc.).
   * **Update Records:** Admission officers can update applicant details, application status, communication logs, and other related data.
4. **User Management:**
   * **Role-Based Access Control:** Different roles (applicant, admission officer, applicant with account) with specific permissions for accessing and modifying data.
   * **Password Management:** Functionality for password change and reset.
5. **Notification System:**
   * **View Notifications:** Users can view system notifications related to their applications or communications.
   * **Notification Logs:** System logs all notifications sent to applicants and users.
6. **Program and Exam Management:**
   * **View Programs:** Applicants and users can view available programs and their details.
   * **Manage Exams:** Admins can create, update, and schedule exams, and applicants can view their exam results.
7. **Dormitory and Scholarship Management:**
   * **Dormitory Requests:** Users can request dormitory accommodation, and admins can view and update the status of these requests.
   * **Scholarship Applications:** Users can apply for scholarships, and admins can manage scholarship requests and statuses.

### 6. Design Principles Followed in the CRM System

1. **Modularity**
   * **Description:** The system is divided into independent modules for different functionalities like User Management, Applicant Management, Program Management, Exam Management, Dormitory Management, and Scholarship Management.
   * **Benefit:** Enhances the maintainability and understandability of the system.
2. **Scalability**
   * **Description:** The architecture includes load balancers and multiple web, application, and database servers.
   * **Benefit:** Allows the system to handle increased load efficiently, supporting growth.
3. **Security**
   * **Description:** Implementation of HTTPS for secure communication, role-based access control (RBAC), data encryption, and regular security audits.
   * **Benefit:** Ensures data protection, integrity, and compliance with industry standards.
4. **Performance**
   * **Description:** The system ensures fast response times with optimized queries, efficient server configurations, and caching mechanisms.
   * **Benefit:** Provides a fast and efficient user experience, handling required transactions per second.
5. **Maintainability**
   * **Description:** Modular architecture, comprehensive documentation, coding standards, and automated testing.
   * **Benefit:** Facilitates easy maintenance, updates, and extension with new features.
6. **Availability**
   * **Description:** Redundant servers, regular backups, disaster recovery plans, and uptime monitoring.
   * **Benefit:** Minimizes downtime, ensuring high availability and quick recovery from failures.
7. **Flexibility**
   * **Description:** Use of RESTful APIs, multi-language support, and configurable settings.
   * **Benefit:** Enables easy adaptation to changing requirements and integration with other systems.