1. **INTRODUCTION**

Student resident information system is an application that is committed to implementing and maintaining technology to foster collaborative learning within residence. It is designed to manage residence activities to help both students and administration in control of managing residence related issues.

1.1 Background and history

The University of Eswatini has been rapidly improving for the past years for better educational infrastructure and environment. Residence is one of the main infrastructures and the university is currently using manual system for residence management which has brought about a lot of challenges.

The current system requires students to travel to campus for accommodation application, form filling for access of materials to use during their stay, having to go to office to report for any hostel related issues, as well as challenges in billing information. This has resulted to poor management of the institution's residence management because there it too much work for the warden which leads some essential students issues not being attended to.

1.2 Aims and Objectives

The main objective of the residence management information system is to automate most of the services to ease/reduce load work for both Residence management office and students residing and/ or students willing to reside on campuses of the University.

With the new automated system students will be able to apply for rooms online and get a feedback to verify if their application were successful. Even those who could not apply in time and are willing to reside on campus will be able to check for availability of rooms and can apply if there is without having to go and confirm to the warden. The system will also have a portal where students can request anything they need after allocation, which may include maintenance services, room transfer, and a lot more of their concerns. It will also enable students to the amount a student owes anytime according to the number of days stayed from the day he/she moved in. There will also be a suggestion page where students can suggest/ or make recommendations (such as security for their safety) during their stay at different hostel blocks.

All these services provided by the system will reduce workload for the warden and also improves the residence management of the University.

1.3 Roles and responsibilities

For the success and completion of the project, all 5 members will be working together. Team work will help in making valid decisions for good quality software that will meet all

stated requirements and solve challenges with the current system. However, each member is has own responsibilities in which the other members may assist on. Bongokuhle Kunene as a team leader, is responsible for overseeing the overall progress of the project. Meetings and other group related matters are looked by her. Mhlengi Mamba is the programmer, responsible for coding. After being given some code contributions by members, he will put together the contributions to come up with a compiling/ running program. Sean Nsibanzde as a designer will look at an overall design of the project, which involves problem solving and planning for software solutions. Kuseni Gumedze is a system analyist responsible for analysing the software using analysis and design techniques using information technology. Hala Mohale is an architect, responsible for high level design choices and frame technical standards.

Each member will focus on assigned role while working with everyone else for a successful software.

2. **User/client involvement**
- Users(students) have to provide adequate information about the current operation of the existing software which mainly includes its deficiencies. A prototype of the existing system can be made and placed in front of the users. Then requirements of the new software system from the users will be gathered. This will be done on the analysis phase of the plan driven approach.
- During testing phase, users will provide feedback on performance of the software and report any errors they discovered.
- During implementation ,the client (university)will notify and train its users on how to use the software with the help of professional technicians provided by the university.
- The university has to provide an IT environment where software will be tested and also provide good hardware where the developed software will run and be tested. Hardware resources has to be made available.
- During testing, the university will check if outcome matches the expected results.
- The university has to hire technicians that will maintain and handle residual errors during normal functioning of the software product to make the software is reliable and healthy.

3. **Risks**

Lack of personnel to perform technical duties- A strategy to avoid this risk could be to recruit skilled personnel to perform required tasks.

Lack of finances- The project needs to run through the employment of resources such as personnel and equipment, all these need money. A strategy to solve this problem would be to seek financing through business proposals or financial loans.

Lack or failure in equipment- Equipment such as computers, commercial software programs or WIFI/internet connection can delay or cause failure to the project. A remedy to solve crushing of computers and networking devices would be to have standby technicians to troubleshoot hardware and network problems. On the issue of lacking in equipment; borrowing equipment from interested stakeholders and donors could be the least expensive route.

Personnel Incompetence- Failure for task members to do their work will result to project failure. Creating a legal instrument to punish incompetent team members could be a strategy to avoid incompetence.

Limited time in completing necessary tasks- Setting a reasonable time frame for tasks could be a useful strategy to avoid not completing tasks on time.

Changing client demands- client may change demands as they wish this could delay or cause the project to fail. A strategy to limit this risk would be early and constant engagement with clients to understand their needs for the project.

TASK (task ID)	Duration (days)	Start date	End date	Predecessor
Introduction(A)	1	11 September	12 September	-
User/Client involvement (B)	6	12 September	18 September	A
Risk Analysis ©	7	18 September	25 September	B
Standards, guidelines and procedures (D)	12	18 September	1 November	C
Project Plan (E)	4	30 September	4 October	C,D
Gathering tools and software (F)	5	30 September	5 October	C,D
Developing UI(G)	12	5 October	17 October	E,F
Implementation (H)	7	17 October	24 October	G
Testing (I)	11	24 October	4 November	H
Client testing(J)	4	4 November	8 November	I
Quality assurance(K)	8	8 November	16 November	j
Project End (L)	1	16 November	17 November	K

ID	Title	Start Time	End Time
1	Introduction	09/11/2022	09/12/2022
5	User/Client invo...	09/11/2022	09/15/2022
6	Risk Analysis	09/16/2022	09/25/2022
13	Standards guide...	09/16/2022	10/01/2022
3	Project plan	09/30/2022	10/04/2022
4	Gathering tools ...	09/30/2022	10/06/2022
7	developing UI	10/05/2022	10/17/2022
8	Implementation	10/17/2022	10/24/2022
9	testing	10/24/2022	11/04/2022
10	client testing	11/04/2022	11/08/2022
11	quality assuranc...	11/08/2022	11/16/2022
12	Project end	11/16/2022	11/17/2022

Timeline headers: Sun 9/04 - Sat 9/10, Sun 9/11 - Sat 9/17, Sun 9/18 - Sat 9/24, Sun 9/25 - Sat 10/01, Sun 10/02 - Sat 10/08, Sun 10/09 - Sat 10/15, Sun 10/16 - Sat 10/22, Sun 10/23 - Sat 10/29, Sun 10/30 - Sat 11/05, Sun 11/06 - Sat 11/12, Sun 11/13 - Sat 11/19, Sun 11/20 - Sat 11/26, Sun 11...