

**FINAL REPORT OF TERM PROJECT**

**CMSE 201**

**Fundamentals of Software Engineering**

**Team members**

**Maas Qureshi, 21904531**

**Saad Ahmad ,22701636**

**Hasan Orkan ,22331350**

**Mustafa Merhametsiz ,22331342**

**PROJECT GROUP : 26**

**GROUP NO : 1**

**PROJECT NAME :Pressure Ulcer Prevention (PUP) OR BED SORE app.**

**PROJECT START DATE :18/10/2023-**

**PROJECT END DATE :12/25/23**

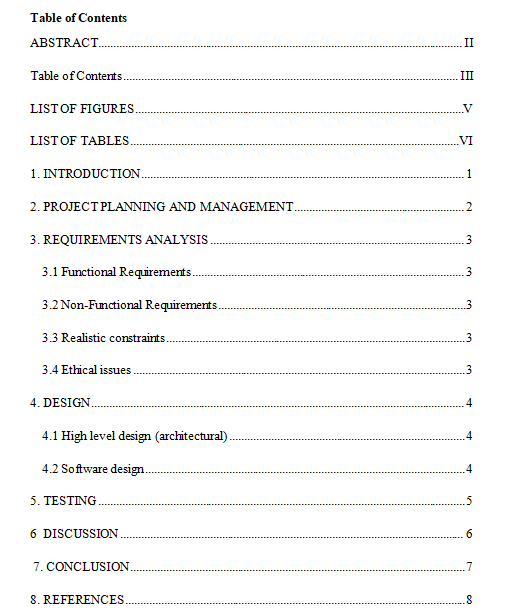
**SUPERVISOR : Prof.Dr Duygu Çelik Ertuğrul,**

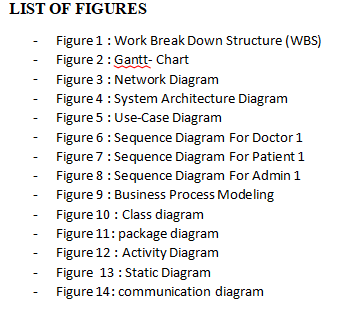
**SEMESTER TERM ::2023-2024 Fall Semester**

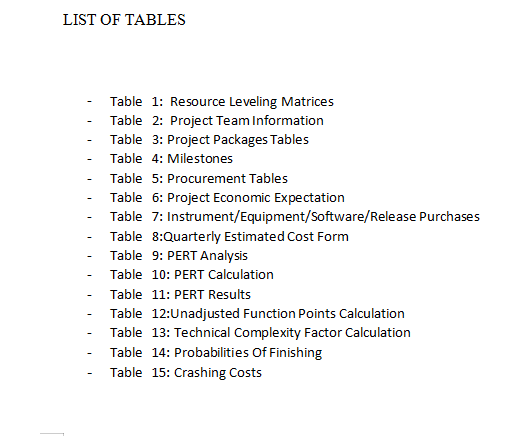
# ABSTRACT

The final report is written to provide an overview and purpose of the pressure ulcer application project that has been developed over a period of 2–3 months. Our team consists of four members. Maaz Qureshi is the project manager and system analyst; Saad Ahmed is the app developer and lead coder; Hasan Orkan is the database developer and administrator; and Mustafa Merhametsiz is the network designer and tester. The application, which is designed to help people who suffer from pressure ulcers or bed sores, monitors their pressure ulcers. It allows patients to track pressure ulcers in high-risk areas, show tips and recommendations for improvement in those areas, and suggest drugs for healing. The principal objective of this project is to develop and deploy a mobile application that offers medical services to patients with pressure ulcer infections who are confined to beds. The application's main purpose is to tell patients who are unable to access healthcare services about their disease sores by scanning or tracking the affected area of their skin and providing a report on the sore's treatment. Our goal is to make the task of accessing health care easier and reduce the cost for patients by recommending decent medications. The project is mainly aimed at the older generation of people who are bedridden or whose movement has become limited. The project cuts down on the process of going to the hospital and waiting there for hours to easily diagnose their disease and get medication at home. For the software planning and analysis stage, we will conduct meetings and interviews with medical experts in the area, as well as do online research and analyze similar systems that perform similar tasks. All scheduling will be documented using MS Project. For the design stage, we will use tools like Modelio to implement the necessary diagrams that will help the development team in the coding stage. For the coding stage, we will use different languages like HTML, CSS, JavaScript, and PHP to implement the requested functionalities and requirements, and for the database system that will keep the user records, we will require some SQL tools like MySQL. For the testing stage, which will be implemented side by side with the coding stage. We will use well-known testing tools to cover more test cases and reach the most efficient time to finish the task. We will also design unique cases observed in the domain that might have catastrophic effects on the application. This technique can decrease the number of patients at hospitals and enhance the number of beds available for other patients, in addition to enhancing the quality of treatment provided to immobile patients. This illness often affects older people, and because they are unable to work or earn much money, the government aids them; thus, this application reduces governments' expectations of financial aid. As a result, health care is not a cheap commodity, and our application allows users to acquire medical treatment at a lesser cost. In the end, we obtained a well-structured planning, SRS, and SDS document from which every stakeholder may benefit and produced a software application that satisfied everything listed in this final report. We consider this project a success since we were able to prepare and present every document and deliverable on time.

**Keywords:** tracking, medicine, drug, diagnosis, prescribed, disease, doctor





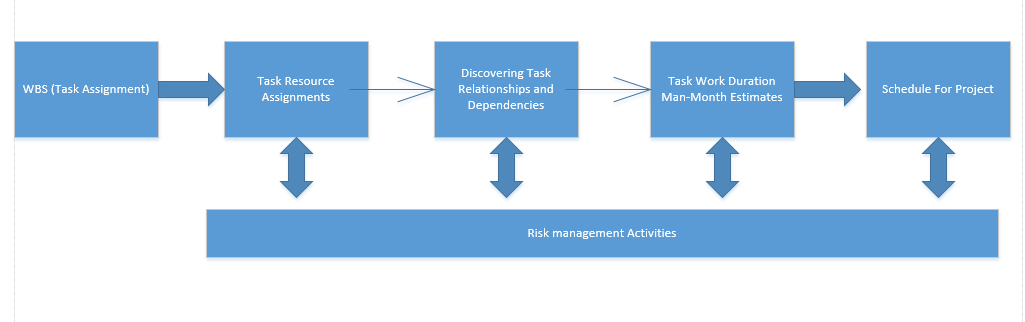


# INTRODUCTION

**The application, which is designed to help people who are suffering from pressure ulcers or bed sores, monitors their pressure ulcers. It sends them daily remainders for pressure relief. This project is based on developing and deploying a mobile application that offers medical services to patients with pressure ulcer disease who are confined to beds. This application's main purpose is to tell patients who are unable to access healthcare services about their disease sore by scanning the affected area of their skin and providing a report on the sore's treatment. It makes the task of accessing health care easier and reduces the cost for the patients by recommending them with decent medications project is aimed at the older generation of people who are bedridden or whose movement has become limited. The project cuts down on the process of going to the hospital and waiting there for hours, just from the comfort of their home. This technique can decrease the number of patients at hospitals and enhance the number of beds available for other patients, in addition to enhancing the quality of treatment provided to immobile patients. This illness often affects older people, and because they are unable to work or earn much money, the government aids them; thus, this application reduces governments' expectations of financial aid. As a result, health care is not a cheap commodity, and our application allows users to acquire medical treatment at a lesser cost.**

# 2. PROJECT PLANNING AND MANAGEMENT

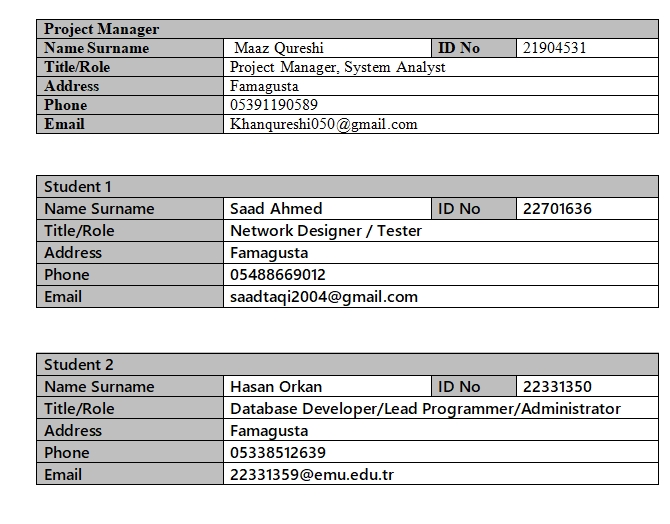
Through whole project, our team walked through the guidelines as well as the life cycle demonstrated in the project management and planning Report. Our execution did not undergo any significant modifications.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role**  **Activity** | **Project Manager/**  **System Analyst** | **Programmer/**  **Tester** | **Database Developer** | **User Interface Designer** |
| **Project Feasibility and pre-Research** | **x** | **x** |  |  |
| **Based system design Technology(Analysis & Design stage)** | **x** | **x** |  | **x** |
| **Development of System Software(Development& stage)** | **x** | **x** | **x** | **x** |
| **Establishing the system Database** | **x** | **x** | **x** |  |
| **Software development** | **x** | **x** | **x** |  |
| **Software Integration** | **x** | **x** |  |  |
| **Prototype Implementation And Maintenance** | **x** | **x** |  | **x** |
| **Project Closure** | **x** | **x** | **x** | **x** |

**Table 1. Resource Leveling Matrices**

**2.1 Preliminary Project Information**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Student 3** | | | |
| **Name Surname** | **Mustafa Merhametsiz** | **ID No** | **22331342** |
| **Title/Role** | **Network Designer / Tester** | | |
| **Address** | **Famagusta** | | |
| **Phone** | **0533 849 69 37** | | |
| **Email** | **22331342@emu.edu.tr** | | |

**2.2 Aim, Purpose and Target Users**

The application, which is designed to help people who suffer from Pressure Ulcers/Bed sores monitor their pressure ulcers. It allows patients to track pressure ulcers in high-risk areas and show tips/ recommendations for improvement of those areas, and suggest them drugs for healing. The principle objective of this project is based on developing and deploying a mobile application that offers medical services to patients with pressure ulcer disease who are confined to beds. This application's main purpose is to tell patients who are unable to access healthcare services about their disease sore by scanning/tracking the affected area of their skin and providing a report on the sore's treatment. Our goal is to make the task of accessing health care easier and reduce the cost for the patients by recommending them with decent medications. Project is mainly aimed at the older generation of people who are bedridden or their movement has become limited. The project cuts down the process of going to the hospital and waiting there for hours, to easily diagnose their disease and get medication in their home. This technique can decrease the number of patients at hospitals and enhance the number of beds available for other patients in addition to enhancing the quality of treatment provided to immobile patients. This illness often affects older people, and because they are unable to work or earn much money, the government aids them; thus, this application reduces governments' expectations of financial aid. As a result, health care is not a cheap commodity, and our application allows users to acquire medical treatment at a lesser cost.

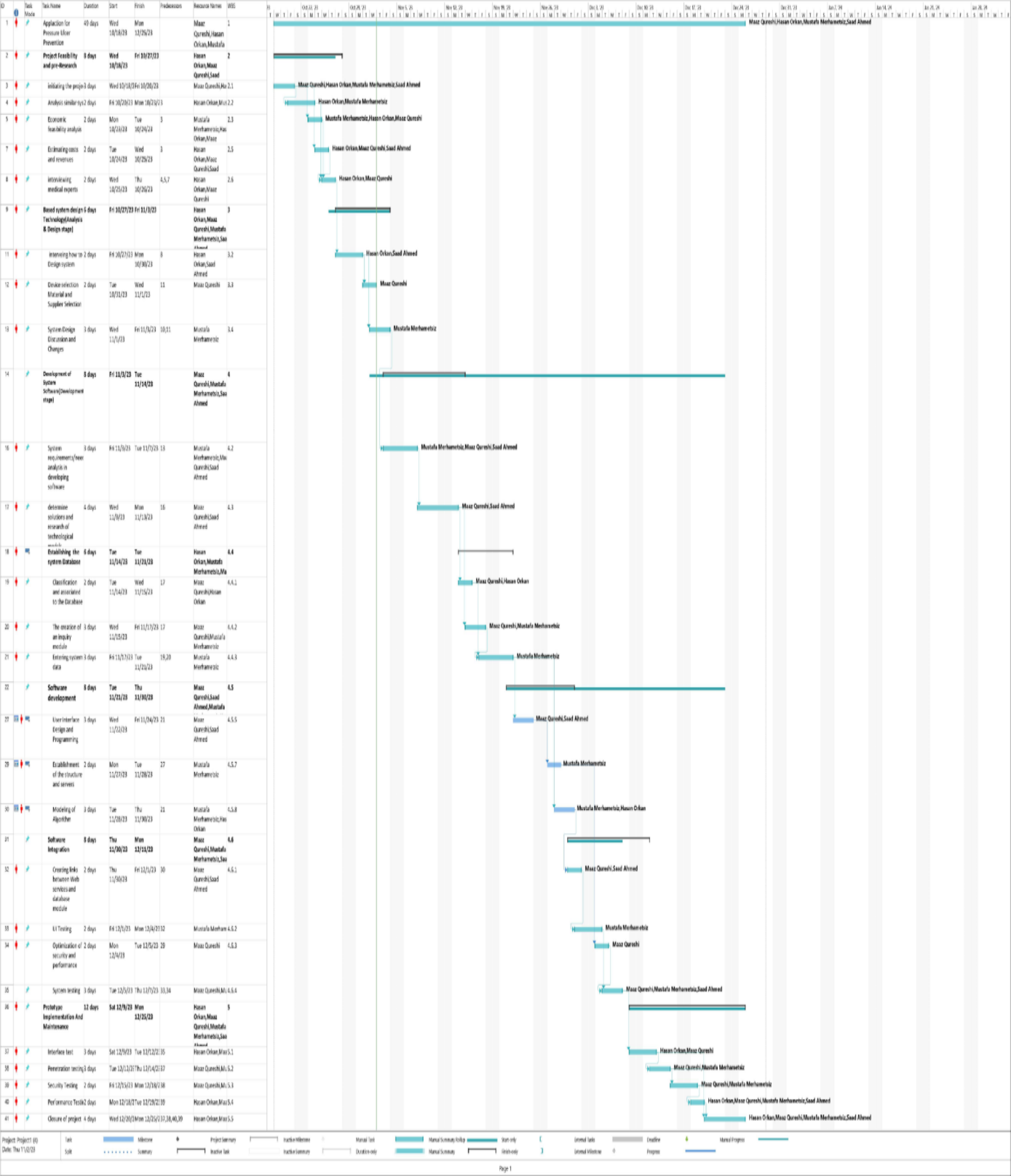
**2.3 The Motivation for Starting This Project**

The main reason for welcoming the beginning of the project is that our team wants to provide bedridden patients with treatment even without a doctor and make them more conscious of their sores. We had a general awareness of the problem domain, and having the chance to implement our knowledge in practical form was an opportunity we did not want to miss. fortunately are no suitable health applications that have camera integration to provide user treatment on the market. Our objective is to develop a user-friendly and feasible tool that employs high-tech techniques to provide for the customers' specific needs, thereby improving their standard of wellness.

**2.4 Output/Success Criteria**

If and only if every user's requirements are fully met, integrated into our system, and displayed in the product's execution, we will consider that our product has been finished successfully and fulfilled its intended purpose. In terms of success criteria, we are considering good sore monitoring and camera integration to detect the stage of ulcer, a good and simple interface for users, good data privacy, a reliable program, and good positive feedback from users. We can tell that our product gives us successful completion and high-quality performance once these concerns are resolved.

**2.2 Gantt Chart**



**2.3 Project Package Tables**

|  |  |
| --- | --- |
| **Work Package No** | **1** |
| **Work Package Name** | **Project Feasibility and pre-Research** |
| **Start-End Date and Time** | **18/10/2023- 27/10/2023** |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| **1.1 Project Process and Economic Feasibility:**  **-In order to improve and streamline the tracking process,we looked into it.**  **-market research to find similar systems and analyze the potential need for this system.**  **-determining how the system works in order to have a correct idea of its advantages and costs.**  **-Analyzing possible risks was evaluated.**  **-We started this project to improve patient care and make it clear for mobile applications.**  **1.2 Technological Feasibility:**  **-Deciding technical and technological requirements needed to make this project.**  **-looking for concept designs**  **-Analysis of software requirements for mobile applications**  **-Patent and academic resource search** |
| **2- Describe the methods and parameters that will be used for work package.** |
| **Technical, legal, academic, functional, and meetings with professionals Ms Project will be used to perform stages.** |
| **3- List the experiments, tests and analysis in the work package.** |
| **-Risk estimations**  **-Technological requirements and user needs’ test**  **- Project process flow test**  **- Selection of experienced group members and deciding on work distribution**  **-Interview with possible users for better requirement specifications** |
| **4- List the output of work package and its success criteria s.** |
| **Outputs:The output of the feasibility study will allow our team members to have a vision of what the project is and understand the general idea of pressure ulcers.**  **Success Criteria:If our system is to track pressure ulcers’ stages correctly, our project will work on it.** |
| **5- Explain the relation of output with other work packages** |
| **This feasibility study package must be completed successfully as it is; it is also valid for our other project packages. If it is completed successfully, we can go on to the next phase without a problem.** |

|  |  |
| --- | --- |
| **Work Package No** | **2** |
| **Work Package Name** | **Based System Design Technology (Analysis & Design stage)** |
| **Start-End Date and Time** | **10/27/23- Tue 11/7/23** |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| **-plan stakeholders for the system**  **-determine the necessary resources**  **-define roles and responsibilities for each stakeholders.**  **-Maintain team members and assign roles**  **-Create visual diagrams** |
| **2- Describe the methods and parameters that will be used for work package.** |
| **-Frequently meeting with all team members**  **-giving tasks for their free time.**  **-Incremental modelling**  **-Making prototypes** |
| **3- List the experiments, tests and analysis in the work package.** |
| **-During the meeting, use all of the members experience**  **-create samples** |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**  **Fully ready system to go to the next stage**  **Success Criteria:**  **If stake holders must be approved by the customer and the system architecture is well designed, can go to the next stage.** |
| **5- Explain the relation of output with other work packages** |
| **Stakeholder approval is important, and a well-designed system is needed for this project.** |

|  |  |
| --- | --- |
| **Work Package No** | **3** |
| **Work Package Name** | **Development of System Software (Development Stage)** |
| **Start-End Date and Time** | **11/7/23- 12/15/2023** |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| **-Design user interface by taken analysis of the project**  **-Create database**  **-Software development**  **-Implement requirements to system**  **- determine solutions and research of technological models**  **- The creation of an inquiry module**  **- Establishment of the structure and servers**  **-links created between database module and web services**  **-User interface and system test** |
| **2- Describe the methods and parameters that will be used for work package.** |
| **-SQLite for database(storing data)**  **-Java to create the main structure of the system**  **-XML for styling the system.**  **- GraphQL to connect our mobile application to database** |
| **3- List the experiments, tests and analysis in the work package.** |
| **- Develop enough space for database**  **-Unit test for strong codes**  **-Interface design and programming with professional tools**  **-Check function-ability of the system**  **-Modelling algorithm**  **-Analysis of system requirements**  **- Optimization of security and performance** |
| **4- List the output of work package and its success criteria s.** |
| **Outputs:**  **- Ready and functional sample of system**  **-Ready for prototype implementation**  **-Ready to see our working program**  **Success Criteria:**  **Well-coded and properly working database well-designed,understandable user interface** |
| **5- Explain the relation of output with other work packages** |
| **In the end, if this stage of the package works well and is done exactly as we were looking**  **Our program's most critical parts are done, and we can see your project.** |

|  |  |
| --- | --- |
| **Work Package No** | **4** |
| **Work Package Name** | **Implementation and Test Study and Maintenance (Test & Maintenance stage)** |
| **Start-End Date and Time** | **12/15/233- 12/25/23** |
| **Related Organizations** |  |

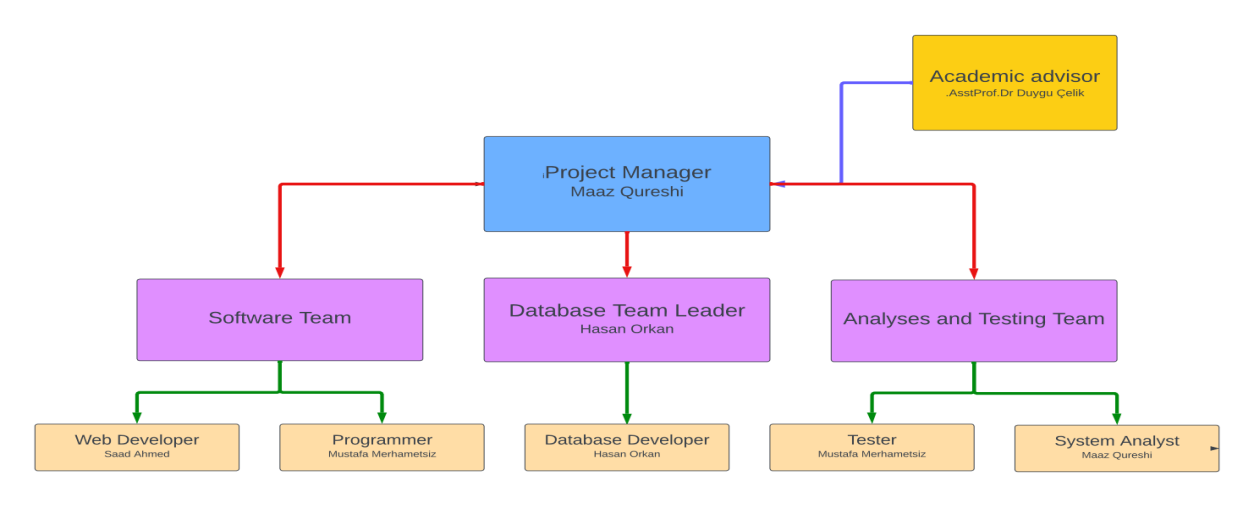
|  |
| --- |
| **1- List the activities of work packages.** |
| **-Security test**  **- Penetration testing**  **-Performance testing**  **-Closure of project** |
| **2- Describe the methods and parameters that will be used for work package.** |
| **- security penetration tests for white-box and black-box**  **- User acceptability testing for requirements**  **- Model-based testing**  **- Unit testing** |
| **3- List the experiments, tests and analysis in the work package.** |
| **-External hacking attacks will be simulated by white-box and black-box**  **- Unit testing will be checking the design and behavior of the applications to make sure that they are**  **-working as we wanted to be**  **-Model-based testing will check the behavior of the system to see if it performing well.** |
| **4- List the output of work package and its success criteria.** |
| **Outputs:**  -Error reports of test  - finished fully working product  **Success Criteria:**  **Minimum error**  **-Trust-able product**  **-Success the aim of the project**  **-Ready to use anytime** |
| **5- Explain the relation of output with other work packages** |
| **If this stage works well without any errors or unwanted behavior, we can end this work package. And this means that we will get what we wanted, so it is almost ready to use.** |

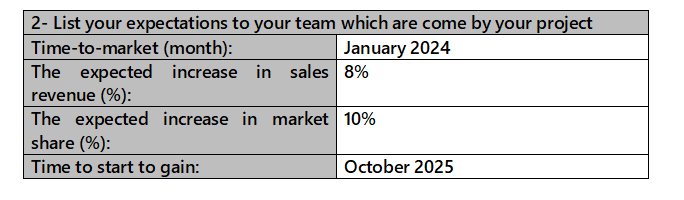
# 2.4 List of Milestones

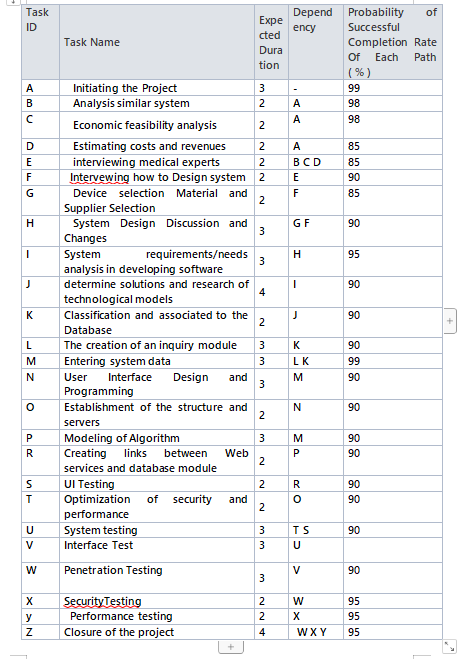
|  |  |  |
| --- | --- | --- |
|  | **Description of Output** | **Expected Time Interval** |
| **1** | **Pre-research and Project feasibility** | **18/10/2023- 27/10/2023** |
| **2** | **Based System Design Technology (Analysis & Design stage)** | **10/27/23- Tue 11/7/23** |
| **3** | **Development of System Software (Development Stage)** | **12/15/233- 12/25/23** |
| **4** | **Implementation and Test Study and Maintenance (Test & Maintenance stage)** | **12/15/233- 12/25/23** |

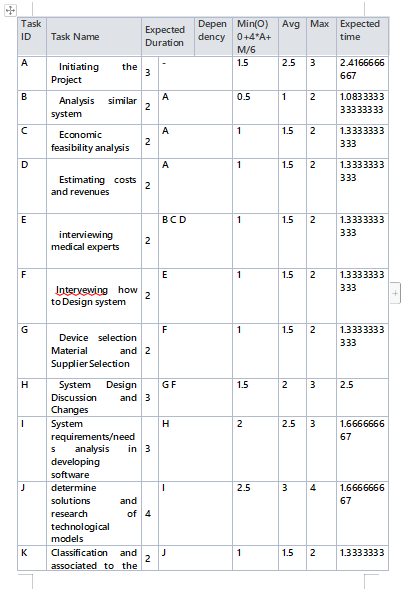
# C.1.4 List of Risks *(see following example, write possible risks for your project!)*

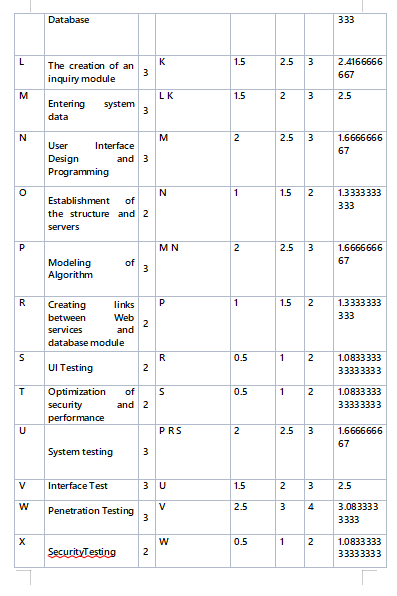
|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Probability** | **Effects** | **Your Strategy** |
| he duration of time required to develop the program is underestimated. | **High** | **Serious** | ***Increasing the frequency of teamwork with meetings*** |
| Software instruments cannot work together in an integrated manner. | **High** | **Tolerable** | **Can find new professional tools** |
| Customers are unaware of the consequences of requirement changes. | **Moderate** | **Tolerable** | **Requirements can change with basic language** |
| The rate of error repair is significantly underestimated. | **Moderate** | **Tolerable** | **Replace potentially defective parts with reliable purchased-in components.** |
| The software's size has been underestimated. | **High** | **Serious** | **Think about buying SW components; consider using a software generator.** |
| **Code generated by code generation tools is inefficient.** | **Moderate** | **Insignificant** | **Code can be written from another tool or language** |
| The system's database is unable to process as many transactions per second as planned. | **Moderate** | **Serious** | **Think about buying a more reliable, higher-performance database.** |
| **Team members might be ill** | **Moderate** | **Serious** | **Work together with other team members until they comeback** |
| **Key team members might leave** | **High** | **Serious** | **Find new team member with** |

****

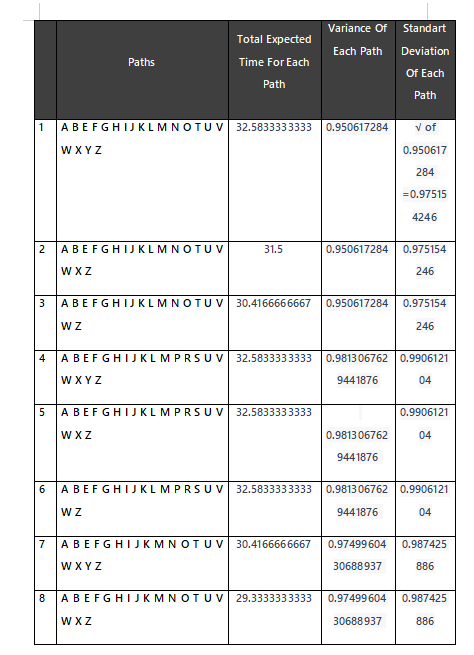


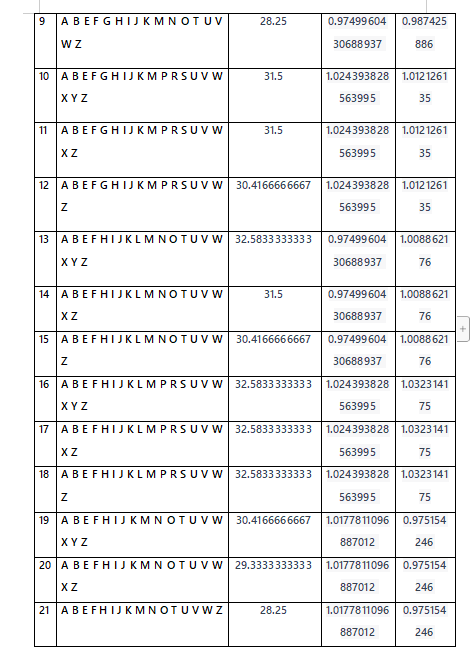
****

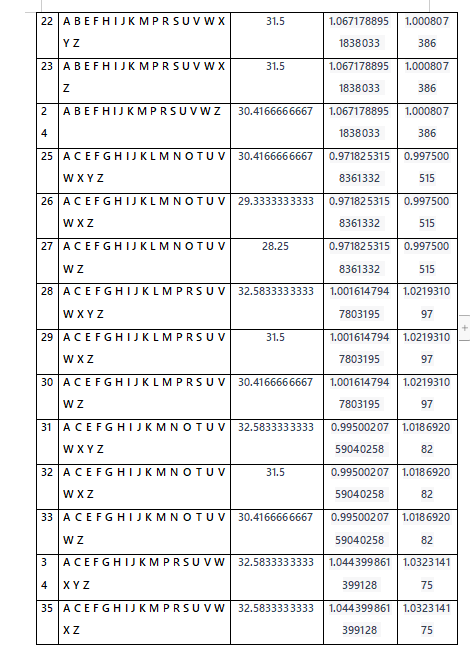
****

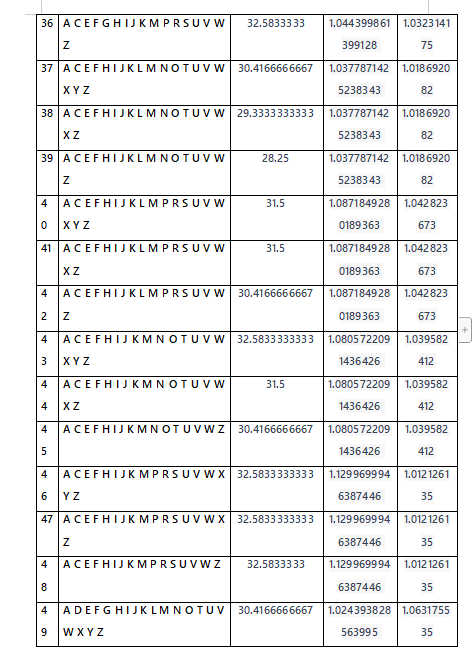
****

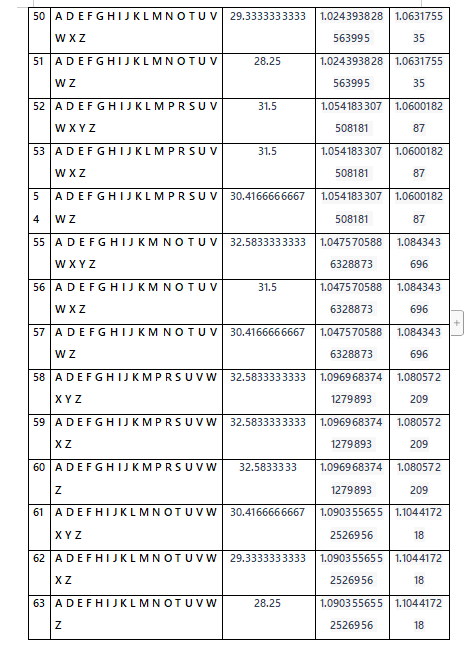
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **y** | **Performance testing** | **2** | **Y** | **2** | **3** | **4** | **3** |
| **Z** | **Closure of the project** | **4** | **W X Y** | **1** | **2** | **3** | **2** |

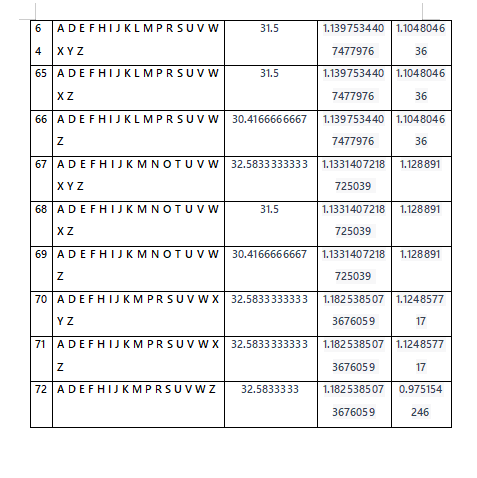
****

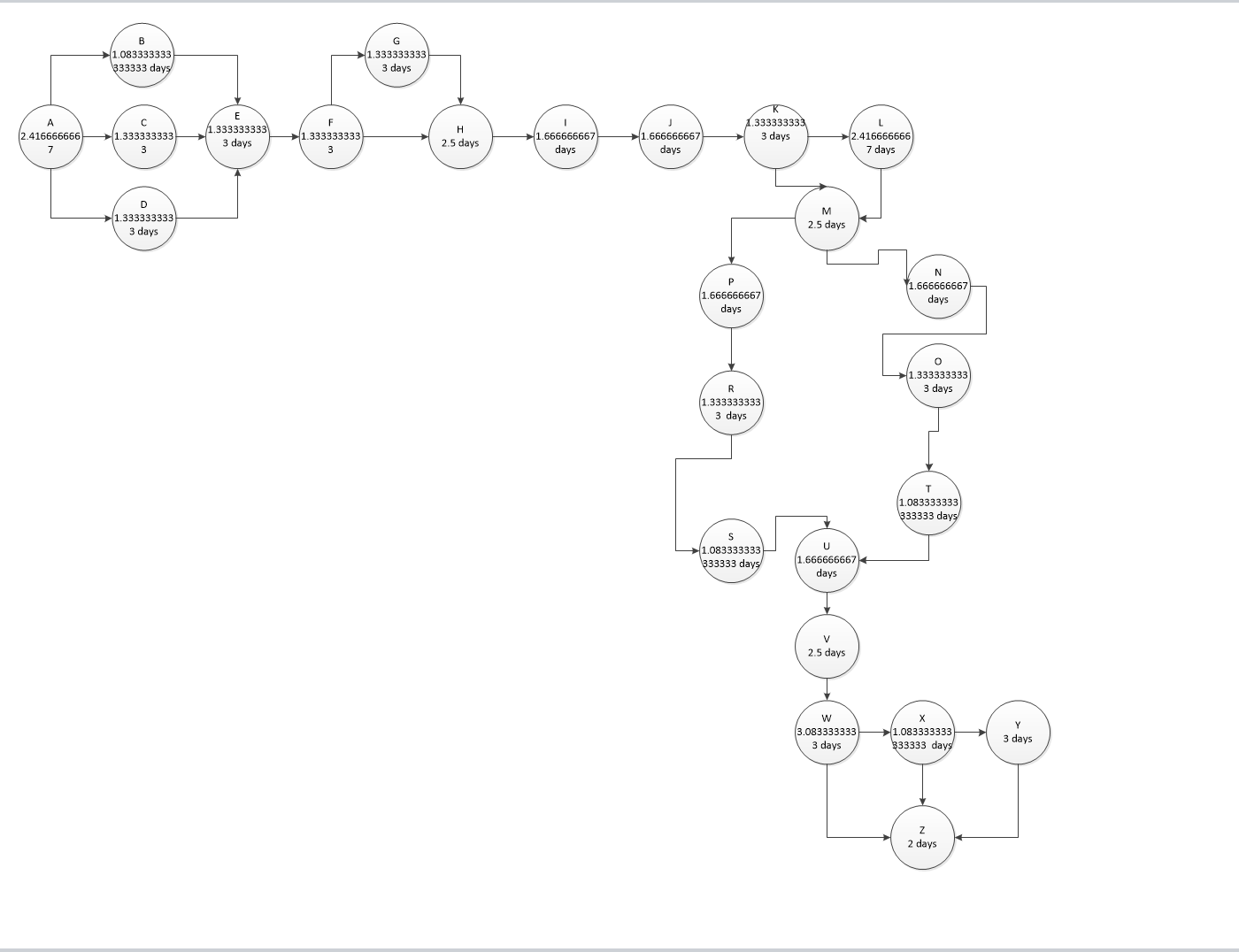
****

****

****

****

****

****

**CPM (Critical Path Management) Analysis By Using PERT (Defining Paths)**

**Once the first path estimation is done, additional paths will be estimated in the the same way by verifying the the table amounts in PERT Calculation Section.**

**PERT CALCULATION**

**Expected time calculated using expected duration values.**

**Variance calculated by using variance values of each process.**

**Standard deviation = taking square root of variance.**

**You can also see all other possible paths.**

**The longest path “A B E F G H I J K L M N O T U V W X Y Z” have 32.5833333333 is our critical path**

**Formulas :**

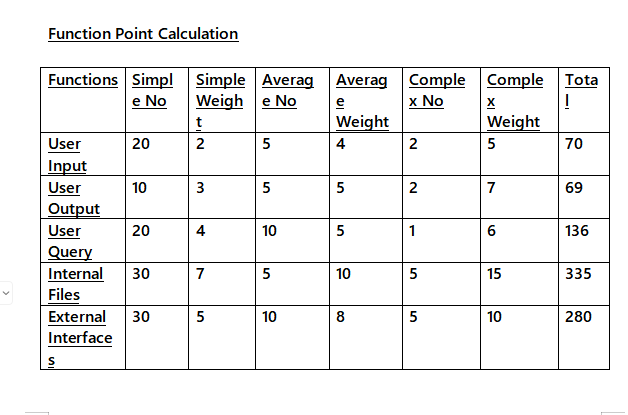
**Expected Time: (min. duration +4\*avg. duration +max. duration)/6**

**Variance:**

**((max. duration−min.duration)/6)^2**

**Derivation:**

**√Variance**

****

**Total: = 70+69+136+335+280= 890 Unadjusted Function Points**

**Technical Complexity Factor (DI)**

|  |  |  |
| --- | --- | --- |
| **Factors** | **Complexity** | **Complexity Value** |
| **Data Communication** | **Essential** | **5** |
| **Performance Criteria** | **Essential** | **5** |
| **Online Data Entry** | **Significant** | **4** |
| **Re-usability** | **Significant** | **4** |
| **Ease of Installation** | **Essential** | **5** |
| **Maintainability** | **Essential** | **5** |
| **Online Updating** | **Significant** | **4** |
| **Complex Calculations** | **Moderate** | **2** |
|  |  | **DI=34** |

**Function Points (FP) = UFP\*(0.65+0.01\*DI)=890\*(0.65+0.01\*34)=881.1**

**COCOMO Estimation**

**KLOC from FP = 881.1\*53=46698.3 LOC =46.6983 KLOC**

**the team is new with the methods and languages used so we will follow an semi-detached type of development:**

**Effort =person-month = a\*(KLOC)^b = 3.0\*= 222.197**

**Total time Duration-in-month = c\*(Effort)^d= 2.5\*=16.56 months**

**Number of Team Members=Effort/Duration= 222.197/16.56= 13.41 person required.**

**Crashing Approach**

**Z= (specified time – expected time) / path standard deviation**

**For each path in critical path management section we will calculate the z value :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **0** | **Expected Time Of The Path** | **Specified Time** | **Standart Deviation Of The Path** | **Z value** | **Probability Of Finishing** |
| **1** | **32.58333333** | **35** | **0.975154246** | **1.586482634** | **0.96983** |
| **2** | **31.5** | **35** | **0.975154246** | **2.697417994** | **0.95089** |
| **3** | **30.41666667** | **35** | **0.975154246** | **4.295078726** | **0.95089** |
| **4** | **32.58333333** | **35** | **0.990612104** | **2.107878854** | **0.95089** |
| **5** | **32.58333333** | **35** | **0.990612104** | **2.107878854** | **0.9879** |
| **6** | **32.58333333** | **35** | **0.990612104** | **2.001742819** | **0.97215** |
| **7** | **30.41666667** | **35** | **0.987425886** | **4.19600033** | **0.97215** |
| **8** | **29.33333333** | **35** | **0.987425886** | **5.293129085** | **0.97215** |
| **9** | **28.25** | **35** | **0.987425886** | **7.088459113** | **0.99403** |
| **10** | **31.5** | **35** | **1.012126135** | **3.877396887** | **0.94602** |
| **11** | **31.5** | **35** | **1.012126135** | **3.877396887** | **0.94602** |
| **12** | **30.41666667** | **35** | **1.012126135** | **4.850523302** | **0.94602** |
| **13** | **32.58333333** | **35** | **1.008862176** | **2.702889346** | **0.98649** |
| **14** | **31.5** | **35** | **1.008862176** | **3.776706324** | **0.97215** |
| **15** | **30.41666667** | **35** | **1.008862176** | **5.535455772** | **0.97215** |
| **16** | **32.58333333** | **35** | **1.032314175** | **1.586482634** | **0.97215** |
| **17** | **32.58333333** | **35** | **1.032314175** | **2.697417994** | **0.96983** |
| **18** | **32.58333333** | **35** | **1.032314175** | **4.295078726** | **0.95089** |
| **19** | **30.41666667** | **35** | **0.975154246** | **2.107878854** | **0.95089** |
| **20** | **29.33333333** | **35** | **0.975154246** | **2.107878854** | **0.95089** |
| **21** | **28.25** | **35** | **0.975154246** | **2.001742819** | **0.9879** |
| **22** | **31.5** | **35** | **1.000807386** | **4.19600033** | **0.97215** |
| **23** | **31.5** | **35** | **1.000807386** | **5.293129085** | **0.97215** |
| **24** | **30.41666667** | **35** | **1.000807386** | **7.088459113** | **0.97215** |
| **25** | **30.41666667** | **35** | **0.997500515** | **3.877396887** | **0.99403** |
| **26** | **29.33333333** | **35** | **0.997500515** | **3.877396887** | **0.94602** |
| **27** | **28.25** | **35** | **0.997500515** | **4.850523302** | **0.94602** |
| **28** | **32.58333333** | **35** | **1.021931097** | **2.702889346** | **0.94602** |
| **29** | **31.5** | **35** | **1.021931097** | **3.776706324** | **0.98649** |
| **30** | **30.41666667** | **35** | **1.021931097** | **5.535455772** | **0.97215** |
| **31** | **32.58333333** | **35** | **1.018692082** | **3.436611525** | **0.97215** |
| **32** | **31.5** | **35** | **1.018692082** | **3.436611525** | **0.97215** |
| **33** | **30.41666667** | **35** | **1.018692082** | **1.586482634** | **0.96983** |
| **34** | **32.58333333** | **35** | **1.032314175** | **3.808353354** | **0.95089** |
| **35** | **32.58333333** | **35** | **1.032314175** | **4.919288714** | **0.95089** |
| **36** | **32.5833333** | **35** | **1.032314175** | **6.772790254** | **0.95089** |
| **37** | **30.41666667** | **35** | **1.018692082** | **3.525412142** | **0.9879** |
| **38** | **29.33333333** | **35** | **1.018692082** | **3.525412142** | **0.97215** |
| **39** | **28.25** | **35** | **1.018692082** | **4.507116829** | **0.97215** |
| **40** | **31.5** | **35** | **1.042823673** | **4.507116829** | **0.97215** |
| **41** | **31.5** | **35** | **1.042823673** | **5.593164723** | **0.99403** |
| **42** | **30.41666667** | **35** | **1.042823673** | **7.356257596** | **0.94602** |
| **43** | **32.58333333** | **35** | **1.039582412** | **3.115919528** | **0.94602** |
| **44** | **31.5** | **35** | **1.039582412** | **4.176004045** | **0.94602** |
| **45** | **30.41666667** | **35** | **1.039582412** | **5.141451765** | **0.98649** |
| **46** | **32.58333333** | **35** | **1.012126135** | **3.01454148** | **0.97215** |
| **47** | **32.58333333** | **35** | **1.012126135** | **4.077996623** | **0.97215** |
| **48** | **32.5833333** | **35** | **1.012126135** | **5.535455772** | **0.97215** |
| **49** | **30.41666667** | **35** | **1.063175535** | **3.436611525** | **0.94602** |
| **50** | **29.33333333** | **35** | **1.063175535** | **3.436611525** | **0.94602** |
| **51** | **28.25** | **35** | **1.063175535** | **3.014541513** | **0.98649** |
| **52** | **31.5** | **35** | **1.060018287** | **5.141451765** | **0.97215** |
| **53** | **31.5** | **35** | **1.060018287** | **6.204906908** | **0.97215** |
| **54** | **30.41666667** | **35** | **1.060018287** | **7.91008947** | **0.97215** |
| **55** | **32.58333333** | **35** | **1.084343696** | **4.793551091** | **0.97215** |
| **56** | **31.5** | **35** | **1.084343696** | **4.793551091** | **0.96983** |
| **57** | **30.41666667** | **35** | **1.084343696** | **5.741457035** | **0.95089** |
| **58** | **32.58333333** | **35** | **1.080572209** | **3.657286852** | **0.95089** |
| **59** | **32.58333333** | **35** | **1.080572209** | **4.699371944** | **0.95089** |
| **60** | **32.5833333** | **35** | **1.080572209** | **4.947750962** | **0.9879** |
| **61** | **30.41666667** | **35** | **1.104417218** | **2.807042812** | **0.97215** |
| **62** | **29.33333333** | **35** | **1.104417218** | **2.807042812** | **0.97215** |
| **63** | **28.25** | **35** | **1.104417218** | **4.35281877** | **0.97215** |
| **64** | **31.5** | **35** | **1.104804636** | **6.390738721** | **0.97215** |
| **65** | **31.5** | **35** | **1.104804636** | **7.409698711** | **0.96983** |
| **66** | **30.41666667** | **35** | **1.104804636** | **8.349516375** | **0.95089** |
| **67** | **32.58333333** | **35** | **1.128891** | **5.283531533** | **0.95089** |
| **68** | **31.5** | **35** | **1.128891** | **5.283531533** | **0.95089** |
| **69** | **30.41666667** | **35** | **1.128891** | **6.949238255** | **0.9879** |
| **70** | **32.58333333** | **35** | **1.124857717** | **4.951101801** | **0.97215** |
| **71** | **32.58333333** | **35** | **1.124857717** | **5.950170028** | **0.97215** |
| **72** | **32.5833333** | **35** | **0.975154246** | **6.851333568** | **0.96983** |

**Probabilities obtained from the z-table.**

**So depending on these values assume we want to reduce our project time by 1 week, then :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tasks** | **Normal Time (day)** | **Crash Time (day)** | **Normal Cost ($)** | **Crash Cost ($)** | **Reduce Cost Per Day ($)** |
| **A** | **1** | **1** | **30** | **35** | **5** |
| **B** | **2** | **1** | **30** | **35** | **5** |
| **C** | **2** | **1** | **35** | **40** | **5** |
| **D** | **3** | **2** | **20** | **15** | **5** |
| **E** | **4** | **3** | **35** | **40** | **5** |
| **F** | **3** | **2** | **40** | **70** | **30** |
| **G** | **2** | **1** | **35** | **40** | **5** |
| **H** | **1** | **1** | **30** | **35** | **5** |
| **I** | **1** | **1** | **30** | **35** | **5** |
| **J** | **6** | **5** | **60** | **80** | **10** |
| **K** | **2** | **1** | **35** | **50** | **5** |
| **L** | **2** | **1** | **30** | **50** | **10** |
| **M** | **2** | **1** | **30** | **35** | **5** |
| **N** | **2** | **1** | **30** | **42** | **4** |
| **O** | **3** | **2** | **40** | **90** | **9** |
| **P** | **3** | **2** | **50** | **90** | **15** |
| **Q** | **11** | **10** | **100** | **160** | **12** |
| **R** | **10** | **9** | **100** | **160** | **10** |
| **S** | **10** | **9** | **100** | **140** | **5** |
| **T** | **5** | **4** | **50** | **58** | **4** |
| **U** | **5** | **4** | **50** | **71** | **7** |
| **V** | **3** | **2** | **40** | **50** | **5** |
| **W** | **2** | **1** | **20** | **60** | **10** |
| **X** | **2** | **1** | **30** | **38** | **4** |
| **Y** | **2** | **1** | **30** | **35** | **5** |
| **Z** | **4** | **3** | **70** | **80** | **5** |

**Normal Total Cost :**

**= 1150 per day**

**= 1150\*7**

**= 8050 per week**

**Chosen activities for reduction : B,C,D,E,F,G,H**

**New Cost : 1150+5+5+5+5+30+5+5=1210 per day = 8470 per week**

**So with additional 30 per day we can recover from a crash of 1 week. (30\*7=210 per week)**

# 3. REQUIREMENTS ANALYSIS

## 3.1 Functional Requirements

**4.1 Patient Feature**

**4.1.1 Description and Priority**

It is possible to receive patient as a customer of the system. He/she first creates an account and then logs in to the system. Upon logging in, the patient will get a menu to choose among the functionalities that he/she have access to. These functionalities will be shown at 4.1.3.

**4.1.2 Stimulus/Response Sequences**

When the user provides a medical history and symptoms of a pressure ulcer, the system will detect its stage and give recommendations for how to prevent it, or the system can suggest drugs to apply to the area; if it is too high and risky, the system will tell the user to contact a doctor.

**4.1.3 Functional Requirements**

REQ-1: Patients should be able to login/logout safely.

REQ-2:Patients should be able to create and manage their own profiles within the system.

REQ-3: Patients should be able to track stage of his/her pressure ulcer with camera integration.

REQ-4: Patients should be able to see his/her history of recorded tracked pressure ulcer.

REQ-5:Patients should be able to receive daily or weekly reminders if he or she wants

REQ-6: Patients should be able to offer educational materials regarding pressure ulcers and preventative therapies..

REQ-7: patients should be able to have a recommendation for a tracked area, such as applying this medicine.

REQ-8:patients will be reminded for taking drugs

REQ-9:patients will be reminded for taking drugs.

REQ-10:**Patients will be** reminded to perform pressure relief

REQ-11:Patients should be able to contact doctor in case high stage of pressure ulcer.

REQ-12: Patients should be able to report errors.

**4.2 Doctor Features**

**4.2.1 Description and Priority**

It is possible to think of a doctor as a medical advisor to the system. As with other users, he or she must first sign up and then log in to the system. After logging in, the doctor will be able to provide the following functionalities in Section 4.2.3:

**4.2.2 Stimulus/Response Sequences**

The system enables doctors to communicate directly with patients when necessary, encouraging an active and cooperative approach toward health care. Similarly, a detailed patient history is available for healthcare professionals to comprehensively grasp the patient’s medical background.

**4.2.3 Functional Requirements**

REQ-1: Doctors should be able to login/logout safely.

REQ-2: Doctors should be able to see the patient’s history of a recorded, tracked pressure ulcer..

REQ-3: Doctors should be able to upload short academic resources.

REQ-4: Doctors should be able to contact the patient in case of high-stage pressure ulcers.

REQ-5: Doctors should be able to report bugs or errors.

**4.3 Admin features**

**4.3.1 Description and Priority**

The system administrator is basically the manager. However, he or she will also perform additional functionalities; there will be no restriction on the system administrator as described in Section 4.4.3.

**4.3.2 Stimulus/Response Sequences**

The user can use the report error area to describe the issue/bug and upload additional screen shots of the mistake/bug in order for the system administrator to read it..

**4.3.3 Functional Requirements**

REQ-1: The system admin must be able to manage all system users.

REQ-2: System admin can modify delete create user accounts.

REQ-3: System admin can get error reports from users and fix them.

REQ-4: System admin must be able to enable regular maintenance mode.

REQ-5:System admin can check academic resources if they are correct

REQ-6: System admin can see all user log in records

REQ-7: System admin should be able to set and alter alert levels for patients.

REQ-8: System admin should be able to configure the system settings..

REQ-9: System admin will be responsible for tracking and enforcing the security and confidentiality of the data process..

REQ-10:System administrators must be able to manage links between external systems  and healthcare databases..

REQ-11:System admin must supply tools for routine maintenance tasks..

REQ-12:System admin must able to updates, backups, and system health checks

**4.4 Features Available For All Users**

**4.4.1 Create Account (sign up)**

4.4.2 Description and Priority

This function is made for users who want to use our program for the first time. By creating an account for our program, the user can store their pressure ulcer stage data and drug history. This is a critical function because almost all other functionalities depend on it and begin with other actions. This is a critical activity, as it is the first operation done by each user. So this feature has a high priority since it is the first thing that all users have to do.

4.4.3 Stimulus/Response Sequences

After downloading and opening our program, the login page will directly show up for the user who wants to create an account or sign in to an existing account. The user must fill in the desired blank details. To register progress, the user must enter detailed information. Once an account is created, the user can sign in with simple information, such as their full name and email. After that, the user can access other features of our program.

4.4.4 Functional Requirements

REQ-1: The system should be able to open for logged-in users.

REQ-2: The system should be able to give error message if any information found same in database while registration progress such as “There is already an account at this email address.”

REQ-3: The system should be able to give an error message for any left blank information such as “You must feel this space!”

REQ-4: The system should be able to send password reset request to user’s phone number or e-mail address if the user clicks reset password option.

**4.5 Update User Information**

**4.5.1 Description and Priority**

This function is made for users who have registered before and want to change or update their user information, such as their email address or phone number. Users must first log in to our program to change or update their personal information. This is a high priority because any error in any field in the user information form will result in accurate advisory services from the website.

**4.5.2 Stimulus/Response Sequences**

After the user logs in to our program and clicks the user personal information option, they can see their detailed information, and if they want to change it, there should be an extra option for changes or updates to the information, such as “edit information." After the user modifies their information, admin must get a notification that the user wants to change their personal information. If admin accepts the modification, it will be done; if not, the system information will be the same and the user will get information about what is wrong.

**4.5.3 Functional Requirements**

REQ-1: The system should have option to modify user personal information.

REQ-2: The system should check rewritten information if it is different or same with the old one.

REQ-3: The system should send error notification to user if rewritten information is same with old one.

REQ-4: The system should send the rewritten information to admin for checking.

REQ-5: The system should return accepted information to the user personal information place.

**4.6 View user Illness history**

**4.6.1 Description and Priority**

As we described, there will be a user personal illness history section where nurses, doctors, or users can see patients’ past illnesses, past tracked pressure ulcers, or past or current drug use. Any user can access the patient's illness history. It will be included in the standard package used for each patient. This is one of the most important features, and the accuracy of all operations depends on the information provided by the patient.

**4.6.2 Stimulus/Response Sequences** When users log in, they can check their own or their patient’s illness history in the History of the Patient section.

**4.6.3 Functional Requirements**

REQ-1: All users should be able to access illness history of patient.

REQ-2: The system should be able to store tracked pressure ulcer data.

REQ-3: Users should be able to enter their past or currently using drugs.

**4.8.Update/Modify User Information**

4.8.1 Description and Priority

This is a feature for users that have accounts within the system. To add new information to the website or edit some old ones, it is necessary for them to provide a document proving the truthfulness of the changes. The changes will be effective until the administrator approves them.

4.8.2 Stimulus/ Response Sequence

Simply click on ‘Modify information’ under ‘My Page’ on your account to change your details after logging in. The system administrator receives an automatic notifications for changes to any settings. Where the admin approves the updates, the data will be changed, or else the system will remain as it was initially.

4.8.3 Functional Requirement

1. The system should enable the user to edit the information.

2.The system should check the information if originating from a higher authority in the organization.

3.The system should revert to the original stage in case the information presented are not valid.

**4.9 Report Errors To System Admin**

4.9.1 Description and Priority

If the user notices any technical problems on the system, they can report them to the administrator, who will correct them. Through user feedback, the application will continue to function well and provide a good experience for everyone.

4.9.2 Stimulus/ Response Sequence

If you login and face any problems, feel free to contact the system administrator. Press the ‘Report errors’ in the corner of the page and explain what the problem is. We appreciate your support, which allows us to enhance this service for everyone.

4.9.3 Functional Requirement

1. When you send an error report, a confirmation message should tell you it's been received successfully. 2. Admins can label error reports as "done" or "being worked on." This helps in tracking reported problems and keeping users posted. 3. Once you submit an error report, the system should follow up with a message. This message verifies that your report has been sent successfully. 4. The system automatically brings in user data like usernames and emails in the error report. This helps the admin solve issues for individual users.

**4.10 Log-out Operation**

4.10.1 Description and Priority

To increase the security of our customers, we made the log-out function available on our mobile application. By clicking the log-out button, users can log out successfully and end their login for our program.

4.10.2 Stimulus/ Response Sequence

Whenever the user wants to end his/her session.

4.10.3 Functional Requirement

1.If system admin confirms operation, the system should clear all the the user's session data.

2.The system should have clear easy logout design.

3.The system should have to warn user before logout.

4.After logging out our system should redirect user to login screen for better quality.

## 3.2 Non-Functional Requirements

## **Performance Requirements**

* Performance requirements for the Pressure Ulcer mobile application are defined to ensure efficient functioning under various conditions.The requirements involve response time, efficiency, execution time, and storage capacity.
* Application should be able to support many users concurrently without any decrease in performance of the application
* Response time of the Application should be less than 2 seconds for user interaction
* Background processes will be optimized to run efficiently without affecting real time user usage and interactions.
* Application will be designed to efficiently manage and store data, ensuring minimal database fragmentation.
* With a maximum round-trip time of 100 milliseconds, the system must maintain minimal network latency.

## **Safety Requirements**

System will implement regular and automated data checks, and backup procedures for user data. It will ensure a quick and reliable recovery process in case of system failure and data loss. System with have regular database optimization processes so system performance is maintained and it does not crash.Application will be implemented with performance monitoring tools to track system performance, identify bottlenecks, and optimize as necessary. Log and monitor user activities for troubleshooting and for protecting the system.

## **Security Requirements**

This section of our report is measures that guarantee access control, maintain reliability of data, promote verification, and manage ability security threats.

* .Application will be integrated with User Authentication and Authorization mechanisms
* .System security has to meet with the demands of the Health Insurance Portability and Accountability Act (HIPAA).
* .The system will have keep backup daily of data
* .Only admin will have ability to view and modify all information
* .The system will remind the user to change their password every 3 months.
* .The system will not accept less than 8 characters without a special key (@$#) for the password.
* .In critical situations, emergency access protocols must be developed.

## **Software Quality Attributes**

**5.4.1 Usability:**

* System must be easy to learn for users, should have a similar user experience to other healthcare applications
* System should be user friendly, well graphically designed easy to use.
* Camera Integration should be consistent to provide the best results for user.
* System should have a tutorial for users to go over every function of the application
* Application should have system uptime up to 99.9% for ensuring reliability.
* Reliable back up and recovery system should be in place.
* Application designed to accommodate any increase in number of users .
* Should be capable of scaling horizontally to accommodate growing number of data records.
* Application should give regular notifications and alerts to both user and healthcare professionals regarding patient treatment plan, condition and any emergency
* Application should be able to provide external resources to patients and collaborate with other healthcare agencies

**5.4.2 Support-ability:**

* Code should be well and follow all coding standards.
* Updates and patches should be regular to maintain application without disrupting normal applications.
* There should be training and support to users, including training materials
* Customer support channels, cater to their problems
* Maintenance of the system should be cost efficient
* Application should be compatible with all types of mobile platforms(IOS,Android)

## **Business Rules**

**5.5.1 Patient Identification:**

Application will assign a unique identifier to each patient when entered into the system. It will ensure accurate identification of patients in the database.

**5.5.2 User Authentication:**

Only healthcare personal who are authorized can see the patients data. This will help protect patients privacy and will ensure compliance with data security standards in the industry.

**5.5.3 Data Accuracy:**

Users will be given the responsibility for maintaining accurate and up to date patient information to ensure reliability of data and to achieve best results as possible.

**5.5.4 Pressure Ulcer Assessment Frequency:**

Regular assessment of the pressure ulcer will be done using the camera to ensure timed identification and monitoring of ulcer to prevent them and correctly diagnose them.

**5.5.5 Severity Classification:**

Application will use standardized classifications to categorize the severity of pressure ulcers using the camera. This will facilitate consistent communication and treatment planning among healthcare personal.

**5.5.6 Treatment Protocols:**

All the treatment plans for pressure ulcer that will be given will be authorized and approved by healthcare professionals and will follow all clinical guidelines and will be evidenced based to ensure safety and security of both patient and healthcare workers.

**5.5.7 Documentation Standards:**

All Pressure ulcer assessments, treatments, and changes in patients status will be documented according to the standardized protocols. This will help maintain a comprehensive and consistent record of the patient for legal, regulatory purposes.

# **Other Requirements**

**6.1 EFFICIENCY**

At least 20 percent of the processor capacity and storage space available to the system shall be unused at peak load

seasonal periods

The system restart cycle must execute completely in less than 60 seconds.

Any interface between a user and the automated system shall have a maximum response time of two seconds.

**6.2 Reliability:**

Application should have system uptime up to 99.9% for ensuring reliability. Reliable back up and recovery system should be in place.

The account update process shall roll back all related updates when any update fails to commit.

**6.3 Scalability:**

Application designed to accommodate any increase in number of users . Should be capable of scaling horizontally to accommodate growing number of data records.

**6.4 Maintainability:**

Code should be well and follow all coding standards. Updates and patches should be regular to maintain application without disrupting normal

applications.

The system shall not be shut down for maintenance more than once in a 24‐hour period.

**6.5 Compatibility:**

Application should be compatible with all types of mobile platforms(IOS,Android)

**6.6 Compliance:**

The application should comply the relevant healthcare standards, Such as HL7 and FHIR.

**6.7 Resource Utilization:**

Application should be optimized to use system resources efficiently, minimizing

effects on device battery and memory.

**6.8 Interoperability:**

Application should be able to integrate with other healthcare systems and databases, allowing for seamless data exchange. And ensure continuity of care, will enhance

the efficiency of healthcare delivery

## 3.3 Realistic constraints

Although this product is a well-designed product intended for the long term usage, there exists some realistic constraints when it is presented to the real world :

* People below age 14 If it is required for them to use it, it should be advised by an adult .
* Patients will contact the doctor only in chance of high stage of pressure ulcers that we can’t give simple treatment. due to ethical issues.
* Every operation will be under control of health government.
* Inadvertently taking over-prescribed medications (such vitamins and minerals) could be harmful to a person's health. That why we give simple drug suggestion.
* The system is very largely reliant on the internet for cases of urgent consultations, searches through academic materials and direct contact with doctors. Lack of access, or traffic bottlenecks could delay the rapid transmission of information and services

## 3.4 Ethical issues

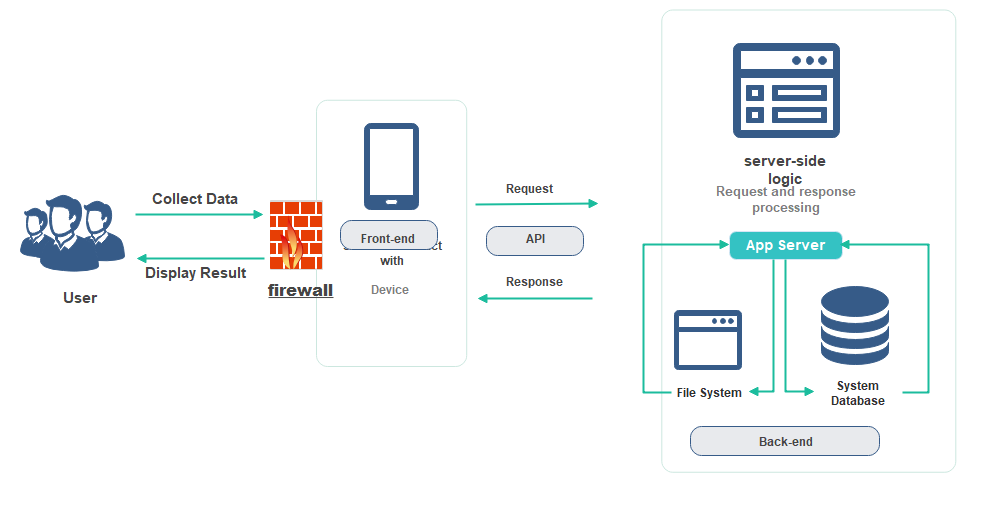
## This technology could be used by those with bad intentions for unauthorized and illegal reasons. The following ethical issues must always be taken into account when using this online application:

* Important ethical issues include determining which parts of conventional research methodologies may be modified for use on the Internet and which parts must be abandoned because they cannot be modified.
* False information regarding illnesses is spread by those who try to manipulate the system and put their own health and safety at risk.
* Online medical services have the potential to skew assessments toward an unduly optimistic view of the benefits of technology by restricting the evaluation of a patient's condition to easily observable or quantifiable elements.
* Researchers are not allowed to use "naturalistic observations" in chat rooms to gather data without the participants' knowledge or consent due to ethical concerns. Informed permission is not implied by merely participating in public forums.
* In the field of medicine, a number of jurisdictions already require out-of-state doctors to hold a license in order to offer patients electronic services.
* More than any other online platform, health-related websites should place a high priority on protecting users' personal privacy. They should ensure that private medical data, usage patterns, and interests are not exchanged, purchased, or unintentionally given to insurers, employers, or marketers.
* There are worries over how some illnesses, like schizophrenia, may affect a person's identity and conduct, especially when it comes to how these problems can show up in online medical services.

1. **DESIGN**

## 4.1 High level design (architectural)

Here in the given system architecture diagram you can see all system components and all external systems as well as their interactions and interfaces.



**The explanation of the main components are presented here :**

**Application server:**The application server is a provides a runtime environment for carrying out applications and enabling communication between multiple parts in a distributed client-server architecture.It manages the transfer of data between clients and servers by using a client-side user interface and a backed database or other services.

**MySQL Database:** MySQL is the most commonly utilized open source database around the world. MySQL has been the chose database choice for App-based apps because of its proven reliability, speed, and ease-of-use, and is used by significant App properties such as Facebook and others.

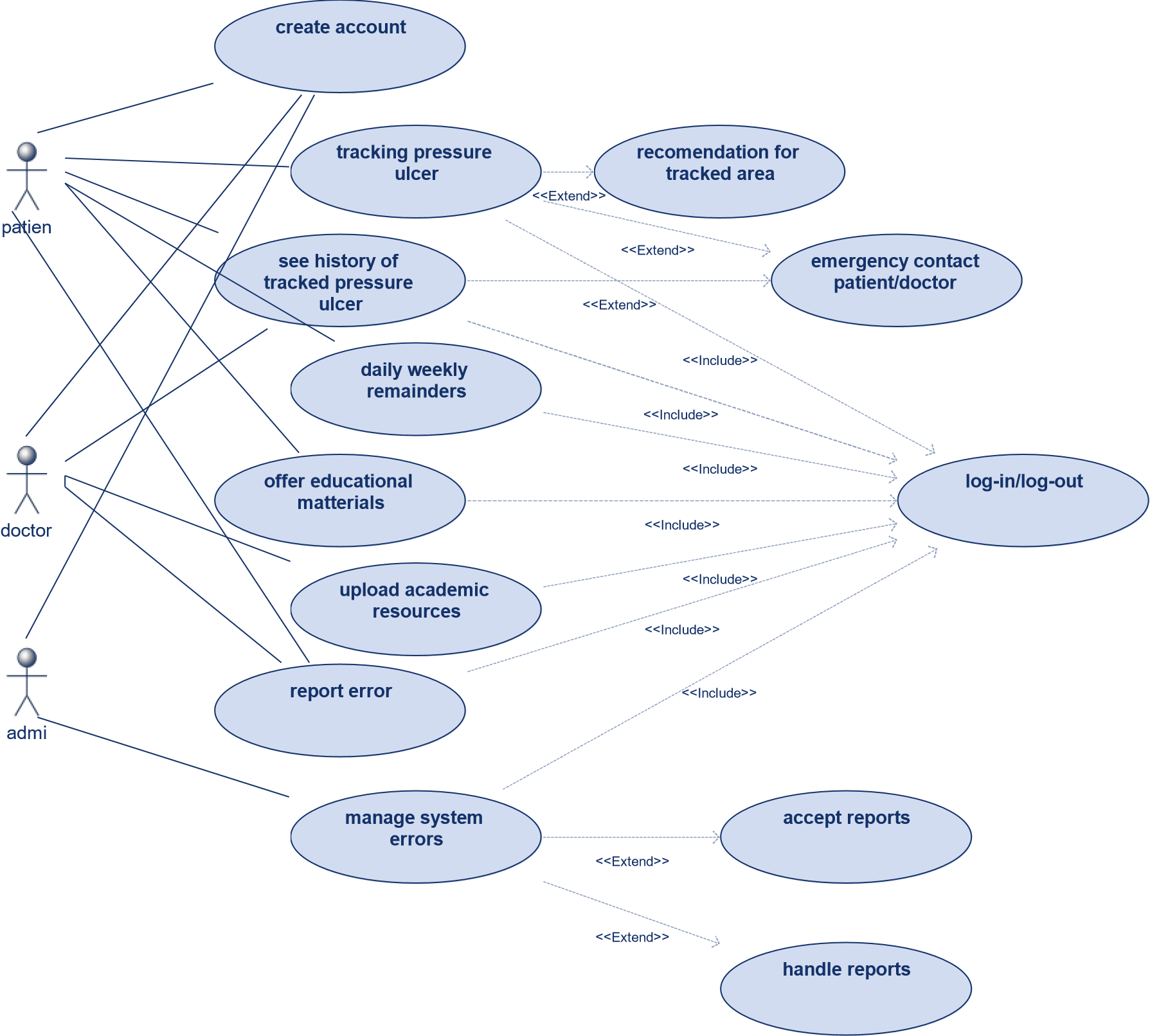
**Server-side Logic:**The generation and distribution of messages and notifications to clients may be managed through server-side logic including optimizations for performance, such as,database i.Sending alerts,Notifying healthcare providers,Notifying healthcare providers

**File System:**The file system refers to an organizational framework or technique that controls, arranges, and stores files and directories on a storage medium. It allows them to create, read, update, and delete files.

## 4.2 Software design

In this section, we will explain our system using UML tools and diagrams in order to offer a more comprehensive overview to our development team and end users who understand and are comfortable with software engineering conditions.

Fig 5 : **Use Case Diagram**

****

A use-case diagram is provided that can be considered a blueprint for our system. It shows core processes and where connection points between actors, namely patients, doctors and administrators both within and outside our system are located. The main processes are intricately linked through two distinct associations: 'Includes' and 'Extends.'

The 'Includes' association means that the process from which the arrow comes out includes in itself the process at its target destination. For example, the "Create Account" feature is something basic that those who don't already have a relationship with the system can use to create an account within it. Except for the 'Register Account' process, very nearly all procedures in our system involve either some sort of Login or are a part of Logout. This deliberate design means that every operation within the system is by an authenticated user, whether a doctor or a patient.

However, the 'Extends' keyword provides the user with flexibility in deciding just how to proceed. In terms of our use-case diagram, patients as well as doctors and administrators can all avail themselves of the "Create Account" feature--a universally applicable doorway into the system. In particular, patients can go through a set of functions spanning the following areas: tracking pressure ulcers; reading their history; receiving prompts (both daily and weekly); getting access to educational material; and reporting errors.

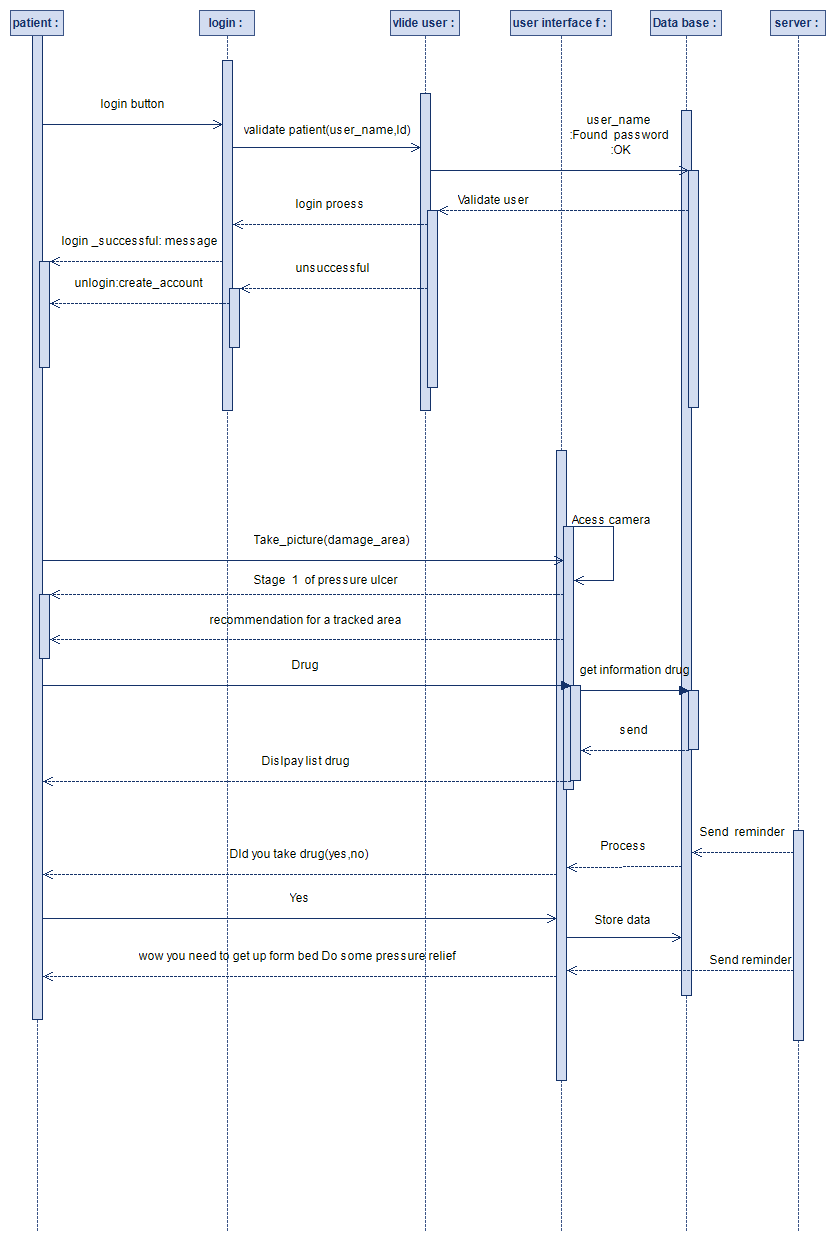
Doctors can use the functionality to submit academic resources online, or get bugs solved in the system. However, only with administrators can they handle all kinds of system errors to maintain overall robustness and reliability.

Moreover, the additions and integration within our functions are arranged in response to special needs and decisions made by stakeholders. The Track Pressure Ulcer function, for example, besides simply monitoring pressure ulcers and their status but also extends its applicability by making suggestions to the tracked area and allowing emergency contact with other users who use it.

To sum up--use-case diagram The use-case diagram presents a bird's-eye view of the system's functions, from the commonality of certain processes, such as "Create Account," to the different features for patients and physicians alike; it is also demonstrative in showing that administrators should take on all of these facets within their lives since they are responsible for Our system architecture is highly user-centered and flexible, reflected in this thoughtful design.

**Sequence Diagrams**

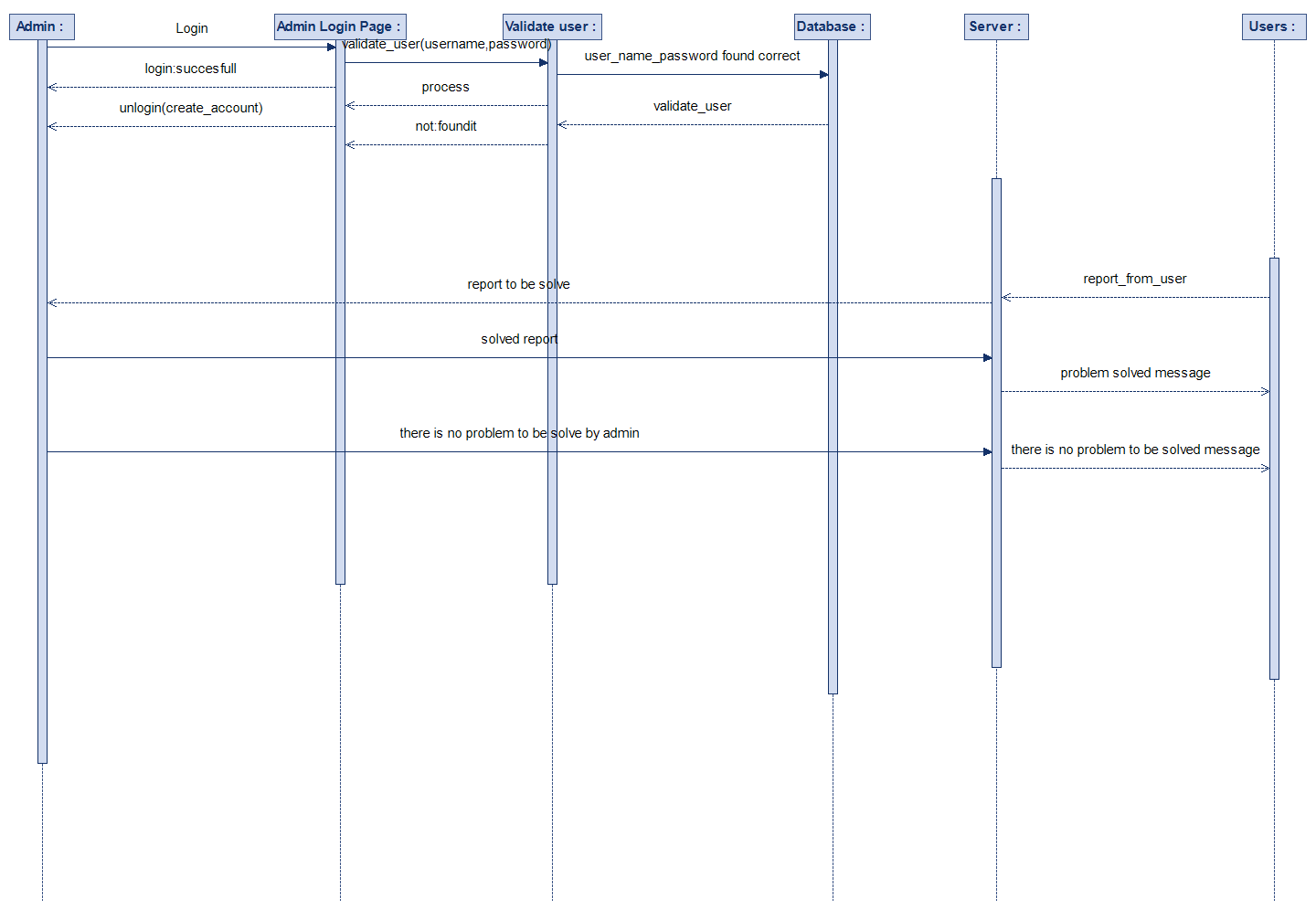
Fig 6 :**For patient**

****

In the diagram above, the patient already has an account, but he must only login first. First, the user will request a login, then fill in the requested information to login to the system. If inputs are incorrect, there will be a message from the system. Firstly, the patient can track their ulcer through a camera, and the server will return to the stage of the pressure ulcer. If the patient wants it, the server will give it, and in emergency situations, the patient can contact the doctor. Patient can report an error to the system, admin can see it on the server, and admin can solve it. In addition, the patient can read the doctor's uploaded academic resources.

Fig 7 :***For Admin***

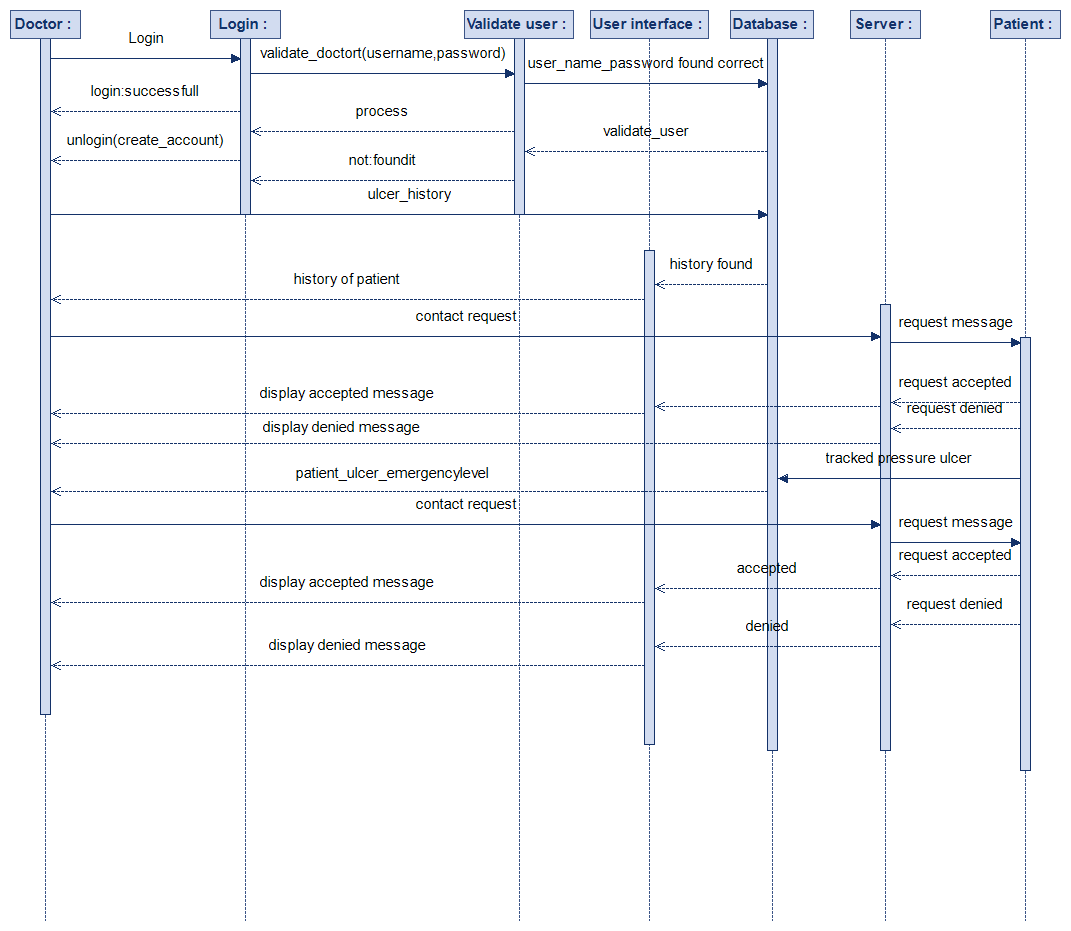
“System admin can get error reports from users and fix them.”



The process depicted in the diagram begins when the admin logs in. Before this they have settled up already an account. The streamlined login process opens up the door to an administrator's vital role in running the system. The administrator systematically examines within the system those error reports that users, whether they are doctors or patients report. This is all part of oversight. When problems aren't found, the administrator quickly takes action to resolve them, further refining and enhancing the system. Conversely, if users report errors but there are no serious problems in need of attention, then the administrator notifies users through messages that such concerns were resolved or simply weren't real challenges. This brief, complete story covers the role of the administrator in preserving the system's stability, ensuring that reported problems are promptly addressed, and keeping good communication with users.

Fig 8 :***For Doctor***

“Doctors should be able to contact the patient in case of high-stage pressure ulcers.”

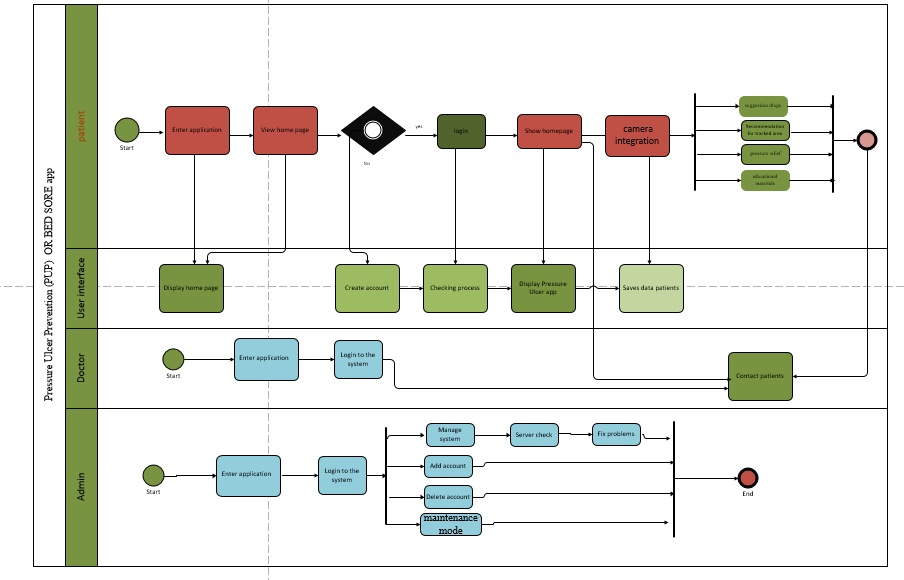


In this example , the process begins with the doctor who already has an account and just needs to perform a login. The procedural flow begins with a user-initiated login request, which triggers the entry of required credentials. If the information entered turns out to be wrong, the system sends a message that forces you to re-enter your data in order to gain entry.

Once logged in, the doctor can get at important features. For example, he can review a patient's pressure ulcer history in its entirety. This access allows a complete examination of the patient's health course, giving insight into their overall state. After all, the doctor who can actively reach out to patients for contact will generate a dynamic exchange of information. The dynamic nature of this interaction goes even further. When a patient's ulcer becomes serious enough to be an emergency, the doctor can reach out and make contact again. In a nutshell, this story describes the streamlined login process which allows doctors easy access to individual patient information; the vast amounts of patient data available to physicians, especially those in emergency contexts.

**Business Process Modeling (BPMN)**

Fig 9 Business Process Modeling



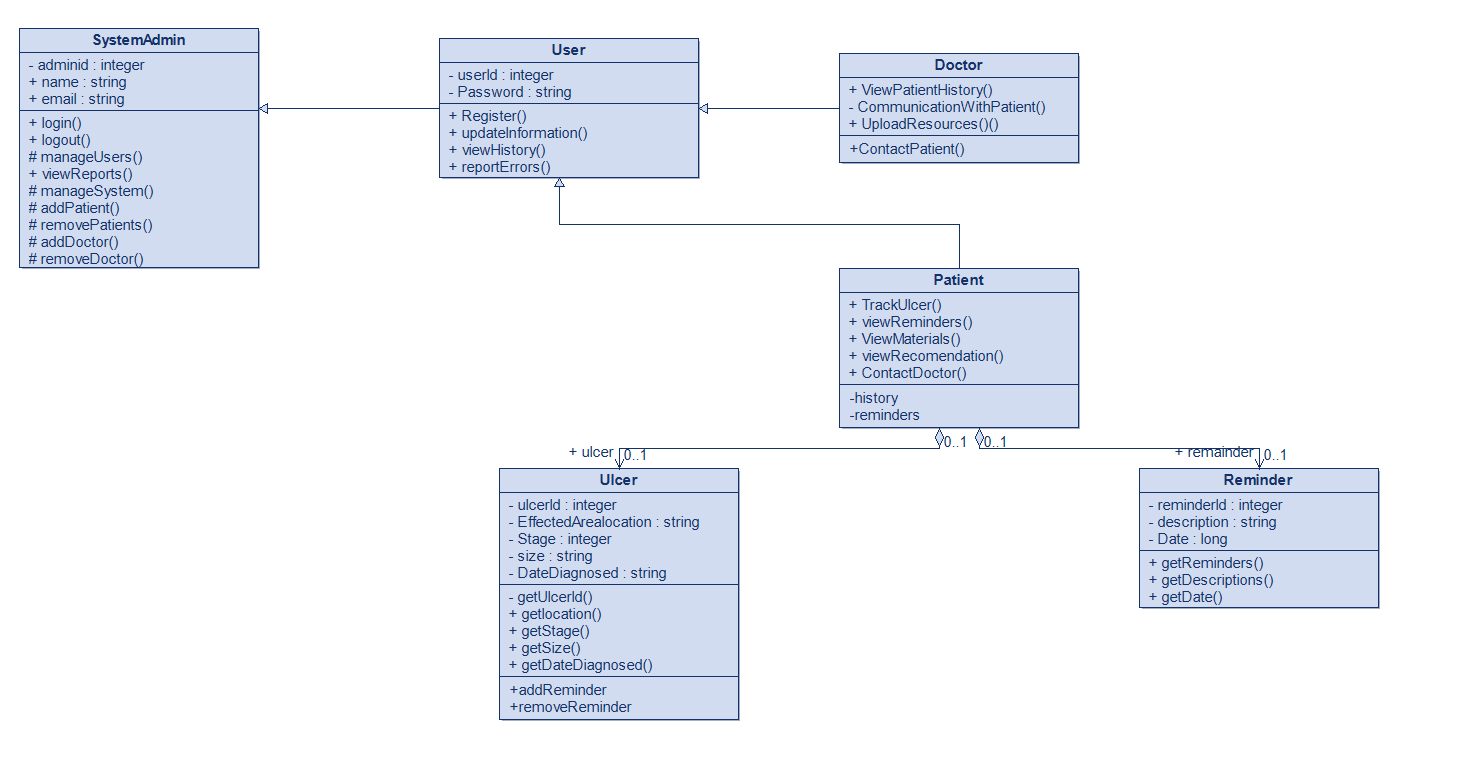
The BPMN Diagram above shows the pressure ulcer treatment app business processes in the diagram. This makes it easier for our technical experts as well as non-technical visitors to better understand the system. It involved three users and three sections in the BPMN diagram. Each section shows how users interact with the system.

For example, a patient will first enter the application; if the patient has an account, it will process; if not, the patient should first make an account; after that, the patient must take a picture of the infected area so the application can give the patient a healing function. The system will give recommendations for tracking areas. other functions.

Doctor: Our doctor can contact our patient if more recommendations are needed. doctors can give advice to our patients.

On the other hand, the admin, who has the highest priority in our system, will be able to conduct all the processes mentioned above in addition to those mentioned above. The admin will manage our system to add or delete accounts, fix problems, etc.

Fig 10 : **Class Diagram**



In the above class diagram, there two parent classes and four child classes. These classes are system admin, User, doctor, patient, ulcer and reminder class. Each class has its own functions and data types operations.

The first class System admin which is the super constructor:

* Can manage all users.
* Can manage the system.
* Remove and add both Patients and Doctor.
* Has login logout function.

System admin basically controls all the system, Manage all the data and users. It is the backbone of the system.

The second class User which is the parent class to doctors and patients:

* Inherits all the available functions from system admin.
* Users are able to control there information and update it.
* Can register.
* Report problems and errors
* View there history.

User registers to the application. Control the information that goes in. After using the application can view there medical history and get treatment. Report an problems or errors to system admin.

The Doctor class is the child class of the two parent classes:

* Inherits all the available functions from system admin and user class respectively.
* Can view patients history.
* Communicate with patient.
* Provide them with valuable resources for the best treatment.

Doctors can login into the system. View their patients, help them with necessary processes and treatment contact patients in case of any emergency.

The Patient class is the child class of the two parent classes:

* Inherits all the available functions from system admin and user class respectively.
* Track ulcers .
* View reminders.
* View recommendations.
* View materials.
* Contact Doctors.

Patients can login into the system. Use the camera integration function to track there ulcers . identify there stages and get correct treatment by using materials resources and recommendations provided to them by the application. In-case of any emergency they can contact their doctor for help.

Ulcer class has an aggregation relationship with patient class:

Contains all the data about the ulcer, its treatment.

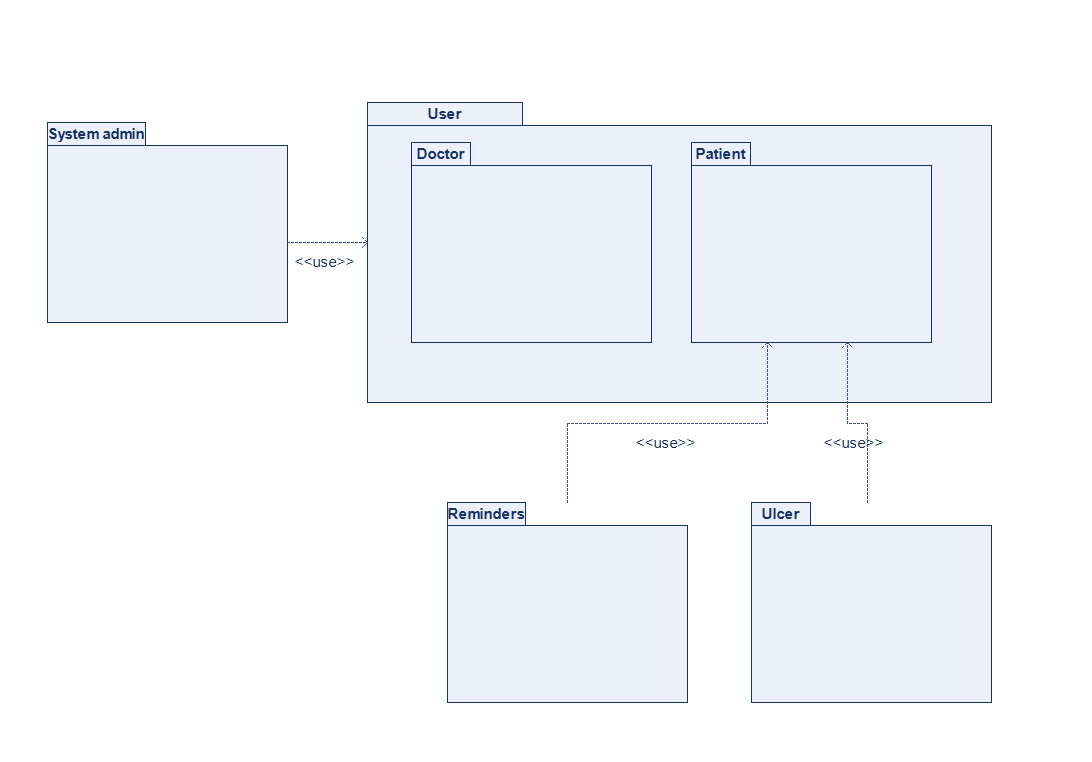
Functions include:

* Get ulcer Id.
* Get location of ulcer on the body.
* Date of Diagnoses .
* Get stage.
* Get Size.

Patient uses the ulcer class to track there ulcers. Being able to track there ulcers accurately to get the best treatment possible.

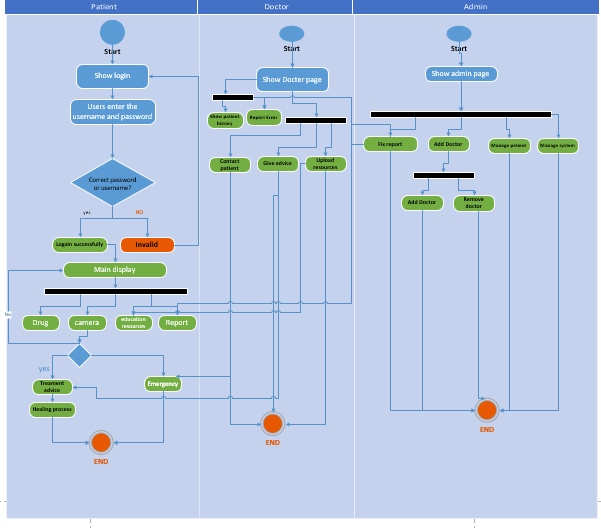
* Reminder class has an aggregation relationship with patient class:
* Get reminders functions for user
* Get descriptions of process and treatment of patients
* Get date of diagnose of ulcer, to get treatment done on time.

Responsible to provide reminders to users to get treatment and other things get done on time

Fig11:**Package diagram**

Above diagram is a package diagram, there are four packages with 2 more packages included in the user package making the total amount of packages to four. System Admin is linked to the user package with last to packages Ulcer and reminder connected to the patient package.

Fig 12 :Activity Diagram:



As shown in the activity diagram, we delve into the dynamic roles and interactions of three pivotal users within our pressure ulcer management system: patients, doctors, and administrators. Our healthcare ecosystem is made up of different users with unique tasks and interrelated functions. Everyone plays their own part in making the system strong and effective.

\*Doctors can upload academic resources.

In emergency situations, the doctor contacts the patient.

\*The doctor inquires about the patient's ulcer history.

If the doctor has already registered, he or she logs into the system. Firstly, the doctors have the ability to upload academic resources, increasing the shared knowledge stored in the system. This joint undertaking improves the medical community's knowledge and experience.

Moreover, doctors can read the ulcer histories of their patients and are engaged in careful observation of those under their care. After they log in, they look through patient records for any abnormalities. In cases of emergency, the doctors' immediate ability to make direct contact with patients makes for a reactive and patient-centered approach to medicine.

Aside from treatment, doctors are involved in proactive system improvement. If, in the course of such interactions, doctors detect bugs or errors, they carefully report their discoveries to the system administrator, kicking off a process of constant refinement.

\*Camera Integration The patient can detect his own ulcer.

Patients can get drug suggestions.

\* Patients can report errors.

If the patient has already been registered, then, after logging in to the system, he or she can use the camera integration function and press on the pressure ulcer for detection. It also provides personalized drug recommendations, giving patients support in maintaining and improving their health. This search function has become a powerful tool. Users can now find specific information customized to their needs.

But it is also patients themselves who are important to the system's improvement.

They can also report errors and actively participate in an environment of constant improvement. In this way, users and administrators jointly create a world where everyone is connected. Additionally, the ability to review one's personal medical history at any time and to respond to a doctor's attempt to contact or initiate contact in high-risk pressure ulcer cases further provides patients with power over their healthcare.

\*Admins can modify user accounts.

Users can report errors to the admin.

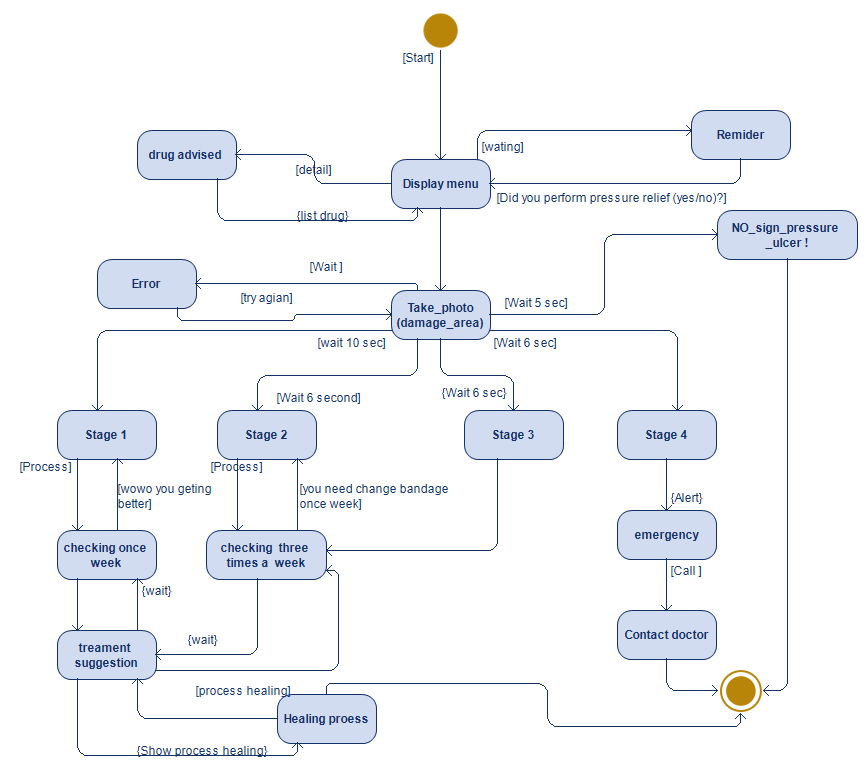
\*To see if it is right, admin can read resources.

When the admin logs into the system, he or she is in fact the referee of our system. Unlike other users, the administrator sidesteps registration and goes straight to logging in to carry out overall tasks.

User account management is one of the vital tasks assigned to the administrator. Administrators, with the power to create, delete, or edit user accounts, keep system security and integrity intact. A key role reveals itself in error correction: as soon as an issue is reported, the administrator fixes it to maintain the system's stability.

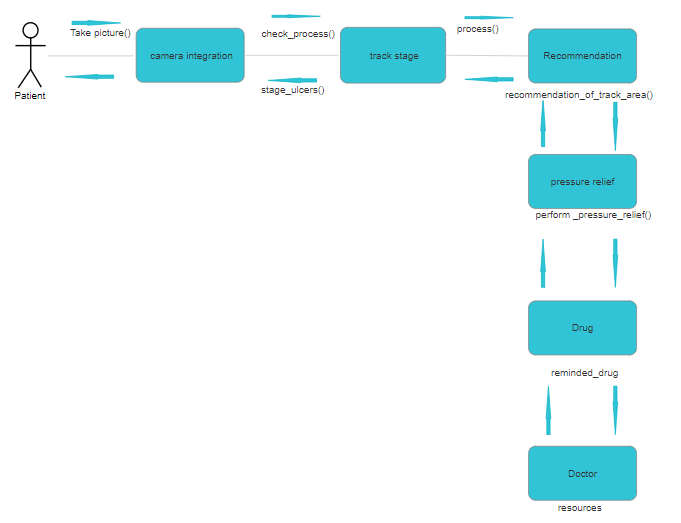
In addition, the administrator reviews error reports from users and does an intensive investigation to find out whether there are any system anomalies. This painstaking care results in an elegant and stable system. Finally, the administrator becomes responsible for vetting academic resources within the system and maintaining a standard of truth and relevance.

Fig 13 :**Static diagram**

****

In the above state diagram, which shows how our system will react to internal and external events, as an example, we show the important reactions of our system. First, we assume that the user, the patient, is already registered with the system. The user enters the password, user name, or email address information to log in to the system. The user will perform a verification process for the account for security purposes. If the user input data is correct, the user logs in to the system and can use our application. Our system will provide drug suggestions to our users, and it will also give constant reminders of: Did he do the given task? For example, pressure relief and taking drugs. Did the patient change the bandage? If they want to access all the services in our system, first the patient should take a picture of the damaged area of the skin, and then the system will determine the stage of the pressure ulcer. If it is a stage 4 pressure ulcer, the patient should contact the doctor. However, if it's stage 1, 2, or 3, we will have treatment suggestions for the healing process of the damaged area.

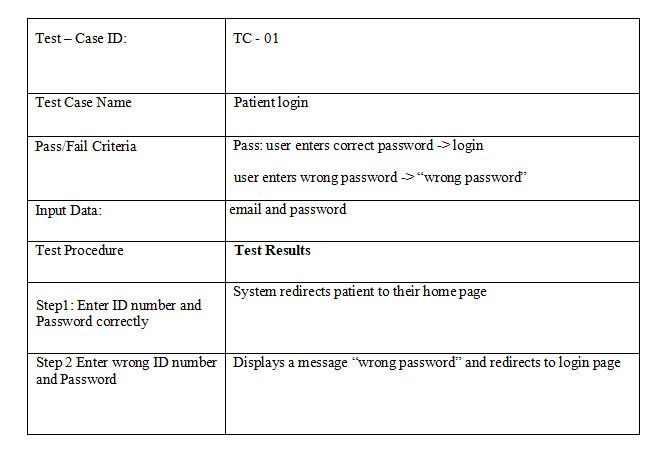
Fig 14 :Communication diagram

.

# 6. TESTING

Our program testing will be mainly done by user testing. As our project uses an agile development model, this process is the best way to test our system, as user testing is involved in this cycle. This testing process will require our input to be exactly like what the customer will do. Tests will be defined by my customers and users. These inputs cannot be replicated in a basic testing environment. This will improve the reliability and robustness of the system. The outputs will be compared to expected outputs and will help improve system databases. During testing, subsequent changes and modifications can be made if there is a case of mismatch.With the help of this, our main features can be tested and improved.

Patient login Test:



Doctor login Test

|  |  |
| --- | --- |
| Test – Case ID: | TC - 02 |
| Test Case Name | Doctor Login |
| Pass/Fail Criteria | Pass: doctors enters correct password -> login  doctor enters wrong password -> “wrong password” |
| Input Data: | email and password |
| Test Procedure | **Test Results** |
| Step1: Enter ID number and Password correctly | System redirects doctor to their main page |
| Step2 Enter wrong ID number and Password | Displays a message “wrong password” and and redirects to login page. |

Patient-Doctor Logout:

|  |  |
| --- | --- |
| Test – Case ID: | TC – 03 |
| Test Case Name | Patient - Doctor logout |
| Pass/Fail Criteria | Patient-Doctor redirected to login page if user pressed yes to logout or stay logged-in  if no |
| Input Data: | Press logout |
| Test Procedure | **Test Results** |
| Step1: press logout | Systems prompt yes or no |
| Step2: press yes | System redirects to login page |
| Step: press no | User stays login |

Update Information:

|  |  |
| --- | --- |
| Test – Case ID: | TC - 04 |
| Test Case Name | Update account information |
| Pass/Fail Criteria | System should change user information based on what they change and save it to database. |
| Input Data: | Name surname email password |
| Test Procedure | **Test Results** |
| Step1: click modify account | System should display page with patient’s information and be able to change their information |
| Step2: press okay | System saves changes |
| Step3: press cancel | System cancels changes and doesn’t save them. |

Camera Integration:

|  |  |
| --- | --- |
| Test – Case ID: | TC – 05 |
| Test Case Name | Camera Integration Test |
| Pass/Fail Criteria | System should track ulcer correctly and identify the right stage |
| Input Data: | Camera scans the effected area |
| Test Procedure | **Test Results** |
| Step1 click on camera button | System opens camera. Scans area |
| Step2 shows Stage | System tracks ulcers and shows stage and treatment |

Reminders:

|  |  |
| --- | --- |
| Test – Case ID: | TC – 06 |
| Test Case Name | Reminder Test |
| Pass/Fail Criteria | System should send Patient reminder according to their schedule and their treatment correctly |
| Input Data: | Treatment and schedule plan selected |
| Test Procedure | **Test Results** |
| Step1 click on treatment plan | System redirects you to a page that shows you treatment plans |
| Step2 select plan | Patients inputs all data needed and select treatment schedule |
| Step 3 send reminders | Systems sends reminder according to the schedule. |

# 8. DISCUSSION

Apart from raising the level of care for bedridden patients, our system may have applicability in a wider field with regard to governmental health programs. This could streamline financial procedures and provide targeted assistance to those who lack the means to even afford basic services. Centralizing health care functions on one platform makes budget management more efficient and transparent. Substituting automated, traceable card transactions for the traditional reliance on cash transactions enhances financial accountability.

Our project aims to democratize health care everywhere in the world, much like with well-established financial concepts such as net present value. Only when this initiative gets to the market will we have the opportunity to work together with other pharmaceutical companies and, as a result, increase our service range, which from a financial perspective may bring very big rewards.

Possible ideas for the project's expansion to a larger organization could involve prepaid care and change how money circulates within the health industry.

Yet there are also doubts about commercialization. For example, who really needs help? Fraudulent prescription cases create even more problems. Even though we already have competitors in the market, our system is different, and a number of its features could be patented. For any work that requires legal factors to be taken into account, partner firms will handle all the arrangements.

# 9. CONCLUSION

In summary, with our mobile pressure ulcer tracking application, we fill the gap between access to care and doctor visits for bedridden patients. It's a big step in that direction. The system will detect pressure ulcers at the user's expense, allowing for speedy intervention by means of individualized treatment plans or medical recommendations. Thanks for that camera integration aspect. The instantaneous contact feature means that urgent discussions with medical experts can take place, so that immediate action will be taken in the event of serious pressure ulcers.

But the incorporation of newspaper articles and other materials that physicians have uploaded enhances patient education by reinforcing self-reliance and making patients informed consumers. This comprehensive approach doesn't just meet the urgent need for healthcare; it also creates a continuous learning environment for both users and caregivers.

Though our goal is to change at-home care for patients who are bedridden, there may still be five or six major restrictions: hardware limitations, problems with internet access, and the necessity of user technological competence. These difficulties highlight the importance of continual improvement and flexibility in an environment of rapidly changing healthcare technology.

Our successful system will be upgraded, taking the favorable effect it has upon patient outcomes as well as insightful user feedback into account. We are committed to protecting data privacy and security, which means that our users 'trust is maintained and regulatory compliance is assured.

In summary, our mobile pressure ulcer tracking device is an innovative new way to provide in-home care, a useful aid both for patients and caregivers on the one hand and ultimately for healthier people and a higher standard of living on the other.

# 10. REFERENCES

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.

IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans.https://www.nhs.uk/conditions/pressure-sores/

Yarkony, Gary M. "Pressure ulcers: a review." Archives of physical medicine and rehabilitation 75.8 (1994): 908-917.