**INDEX**

**Topic Page**

**1. Introduction 2**

1.1. Purpose of the project 2

1.2. Scope of the project (Importance and Challenges) 2

1.3. Motivation of the project (Overcoming Existing Limitations) 3

1.4. Overview of the document

**2. Background Study & Project Planning 4**

2.1. Background Knowledge 4

2.2. Related Works [If remains] 4

2.3. Plan to Develop the System 5

2.4. Risks and Contingencies 5

2.5. Software Cost Estimation 6

**3. Requirements Specification 9**

3.1. Functional requirements 9

3.2. Non-functional requirements 9

**4. Design Procedure 10**

4.1. Architectural design 10

4.1.1. Block Diagram 10

4.1.2. Block Diagram Description 11

4.1.3. Use-Case Diagram 12

4.1.4. Use-Case Diagram Description 13

4.1.5. Class Diagram 15

4.1.6. Class Diagram Description 16

4.2. Database Design 17

4.2.1. Extended ER Diagram 17

4.2.2. Schema Diagram 19

4.3. User Interface 21

4.4 Tools Used 22

**5. Testing and Results 22**

5.1. Test plan overview 22

5.2. Test- procedure with Sample data [Snapshot of Collected Data]

5.3. Performance Analysis 30

**6. Conclusions 31**

6.1. Summary 31

6.2. Discussions on methodologies and potential impacts 31

6.3. Limitations & Future Works 32

**1. Introduction**

In this modern time of ours technology is being used in almost every field such as military field, banking sector, business sector, medical sectors, etc. Medical sector uses various modern machineries but there no software solutions yet that may help any general doctor in analyzing their patient, such doctor’s aid software are used in many developed countries these days. Our aim is provide a medical aid software that will assist a doctor to provide a more correct treatment for their patient. MedAid+ is a software which will provide a new dimension in our medical sector.

**1.1 Purpose of the project**

MedAid+ will play a key role in helping doctors all over Bangladesh and will help to take the medical sector of Bangladesh to next level, as our software will not assist doctors but will connect all the doctors’ research in a single platform so that any doctors can view latest research that other doctors are doing.

* MedAid+ will provide actual case studies or patient records to the users which is constantly updated by doctors.
* MedAid+ keeps all its patient’s records and case studies in a centralized data base system which can be accessed by doctors from other hospitals for reference purpose.
* Researchers will also be benefited by this software system.

**1.2 Scope**

* User target includes all doctors and researchers in Bangladesh.
* Doctors and researchers will get username and password to use this software
* Doctors can use a symptom checker to help in making accurate decisions which uses graphical user interface and is user friendly.
* Doctors can record patient’s record such as name, age, sex, height, weight, diagnosed diseases, location, date in a centralized database system of MedAid+
* Doctors can record patient study cases in a centralized database system which can be linked up with the symptom checker and hence can be searched using symptom checker.
* Patient record system over a large span of time will have gathered large amount of data regarding diseases.
* Researchers will have access to these data collected by patient record system which will help in epidemic prediction and flow.

**1.3 Motivation**

Few years back there was an incidence where an old man in his sixties had experienced sudden difficulty to speak, so his family took him immediately to a clinic. There a young doctor probably an intern was only available at that moment. The young doctor thought the old man was suffering from cold and prescribed medicines for cold and sent them home. Later that night that old man died. It was found out later that he died of stroke. The members of the family was furious about the mistreatment given by the doctor. Mistakes like these are often heard in the news these days, which could have been avoid had the doctor been more careful or had some other doctors to consult with. It was these incidents that inspired us to develop MedAid+ in order to assist doctors in diagnosing their patients.

While making this software we also faced several limitations mostly technical in the developing phase of the software while we have also foreseen future limitation that may arise after the software is developed. Firstly getting accurate data for our symptom checker was a challenge but thanks to webMd we were able gain information about human body parts, many diseases and symptoms, and how the symptoms were linked with various diseases. Second challenge was designing the databases such that various body parts and their related symptoms and symptoms that are to a diseases were given proper relations. While designing we faced a limitation of how symptoms were linked up with various diseases, as a disease have several symptoms and also a symptom can be present in multiple diseases.

**1.4. Overview of the document**

The following document contains Background Study & Project Planning, Requirements Specification, Design Procedure, Testing and Results and Conclusions for the software MedAid+.

**2. Background Study & Project Planning:**

**2.1 Background Knowledge:**

There are certain knowledge we need to have to develop this project. As this product is gives a huge help to doctors specially to our country so we had to ensure all doctor can use it with ease so we chose cross platform software which runs on windows and Linux for that we needed to know how software that we used (Netbeans) works. We needed good knowledge on JAVA as its base language is java, moreover we needed to have good knowledge on JAVA SWING. These were our Implement part background knowledge now for MedAid+ we needed to know how many disease available for initial database, how it works, what can be certain possible symptom for a certain disease, Moreover we needed to know all body parts so that we can add them on our software as different body part or sub body part in Symptom checking. In addition, he had to understand how to make the software user friendly to doctors so that they can understand easily and use them properly.

To conclude, we to know different perspective of background knowledge for example, Implementation coding knowledge such as Java, Java Swing, Netbeans, how system will work, how system will be better and user friendly to doctors also about disease, symptoms how common they are and different body part of human.

**2.2 Related Works:**

There are several existing systems which provides some features that our new system has to offer. We like to mention some existing systems that has some similar features like us. They are mentioned below:

**WebMD:** WebMD is primarily known for its public website, which has information regarding health and health care, including a symptom checklist, pharmacy information, drug information, blogs of physicians which specific topics and providing a place to store personal medical information. Symptom checker is the one feature that Web MD provide like us which is available online for everyone. WebMD symptom and disease database is our primary source of database.

**Electronic Medical Assistant**: This is an electronic medical record (EMR) system specific to diagnostic practices. Modernizing Medicine supports quality of care reporting and outcome-based reimbursement. EMA was designed with structured data technology that handles the Value-based Payment Modifier and enables quality reporting, including Physician Quality Reporting System (PQRS) program reporting. The system can also remember the specific requirements of each provider through an adaptive learning engine, and physicians are able to actively select commonly used diagnoses, prescriptions and consent forms.

This were related works that we found on internet where WebMD is totally free to use but EMR charges money on time subscription. Our system is close to that but it has more features then both of the software combined has.

**2.3 Plan To Develop The System:**

Our initial idea to develop this project will be break into certain category. They are,

* Cross platform software
* Using Java Swing language code
* Database host Selection
* Database will have some initial data
* Making user friendly UI

Now to develop this software we planned to make this software available on cross platform such as Windows and Linux So that any pc user can have access over the software anytime he needs. In addition, as we are planning to make this software cross platform so we planned to use java swing or java as our implementation code. Database selection plays a huge part in this system because as we are planning to keep huge amount of data and files of research so normal database will ran out of data fast so we thought about implementing one Java oracle Enterprise edition database and later it can upgrade its size by Oracle. Oracle Enterprise is popular among Industrial usage of database so we though to use this in our system as well. Now MedAid+ software will be much accurate to use when it will have huge data that stack up by doctors over the time as it will need some time to populate data so we though to use WebMD’s free database data to use initially but when Doctors will use them it will change data and also give better approximation over the time. Lastly, User friendly UI is one of the most important thing on a software as this software will be used by Doctors so it should be friendly to them so we searched for some idea and tried to make it as optimal as it is possible So that, this way this product get better response from doctors.

**2.4. Risks and Contingencies:**

MedAid+ is unique software in medical software. In Bangladesh this kind of software is not available or don’t have as much features as us and few companies thinking so largely because our client will be direct Govt or Privet medical institutes. If we Divide Risks of the project in three categories and they are,

* Project Risk
* Product Risk
* Business Risk

**2.4.1 Project Risk:**

There can be Project risk as our team is small so it can affect schedule of project because less people won’t be as fast as large group of people and our resource is enough but no surety that it will be enough all the time so there is obviously a project risk.

To avoid this kind of risk we need to be prepared about Possible Resource need for future and as our team is small so all project member either has to work overtime which may affect product risk else we need to increase our team size so that we don’t face any schedule problem or resource problem.

**2.4.2 Product Risk:**

There is some product risk in this software. As our team is small it might face project risk so if they work overtime then even if they can solve project risk but it can affect the quality of software so there will be product risk and also it might will effect on its performance (ex: software gets slow after loading database) etc. can occur.

To avoid this kind of risk we need to increase our team size so that we don’t need to have anyone’s overtime and further more Project manager can moderate the programmer’s progress. This way MedAid+ will be free from product risk because if one programmer doing code for 2hr rather than 8hr then, 4 programmer’s 2hr will be more productive and better than 1 programmer’s 8hr even if moderation of the project will enhance performance and quality. Therefore we can say this way we can ensure products quality and performance up to mark.

**2.4.3 Business Risk:**

So if we look closely no companies in Bangladesh currently give this kind of software support so there will not be any business risk. Even if some company make a new software slidely better than us but the data we will have on that time will be much more important that their software and also we can upgrade our system as we followed version.

This were the risk could possibly arise. We gave specific solution to some specific problems but some unknown risks we will follow risk management process which will have,

* Risk Identification
* Risk Analysis
* Risk Planning
* Risk Monitoring

**2.4.4 Risk Identification:**

We will look at the all possible risks for MedAid+ and identify them, here all risk can be any of Project, Product or Business Risks.

**2.4.5 Risk Analysis:**

Here we will look at the prioritized Risks for MedAid+. For Example: Suppose tomorrow is last day to submit the project diagram to get approval but we delayed as our main designer is sick so without this project will be on project risk, here some priority cases like this will be listed and order will be top priority to low priority.

**2.4.6 Risk Planning:**

We will make some plan so that this risk will be reduce for MedAid+, For Example: Moderation by a expertise to check the efficiency of code will enhance the performance of software and won’t let product risk to happen so this way we can avoid by making or drawing a way out of it.



**Fig 2.4.1: Risk Management for MedAid+ software**

**2.4.7 Risk Monitoring:**

We will monitor all the time to see if any existing risk has any possibilities to reoccur or not this way old risk will be monitored for MedAid+ system.

**2.5 Software Cost Estimation:**

Software Cost is predicting the resources required for a software development process , now our rough software Cost Estimation is,

|  |  |
| --- | --- |
| **Development Cost** | |
| Oracle Database Enterprise Edition | $33,250.00 |
| 2 Servers | $7,660.00 |
| Software Update License | $10,010.00 |
| Software Update Support | $8,800.00 |
| Office Space | $5,120.00 |
| Equipment’s- | $4,225.00 |
| Development Labor | $88,000.00 |
| **Total Development Cost** | **$157,065.00** |

**Table 2.5.1: Development Cost**

Now, from Table 2.5.1 we can see the possible Development cost we need. As we said earlier we need Oracle Database Enterprise Edition as MedAid+ software’s data size is huge and will need Industrial usable database we are going to use Oracle Database Enterprise Edition which going to take around $33,250 almost. Then we plan to use 2 Servers because even if one has problem but other one can give backup mean time we can fix the bugged server. That is going to take $7660.

We going to need software Update License and Update Support of database which going to take $10,010 and $8,800 respectively. Office space and Equipment’s are also going to need to do this project so we planning to take $5,120 & $4,225 respectively.

Now Development Labor has different way to fix one employee’s salary we will look to Object Point’s and final Code size to estimate their salary. We will also look into their productivity level to give salary. We assumed in total Development labor will be around $88,000. So in total

We going to cost almost $157,065 so it’s simple we need to charge more than our cost.

As it follows COST ESTIMATION UNCERTAINITY so we are going to charge a bit more than our cost to ensure we stay in profit.

To conclude, MedAid+ will have huge amount of data so components of this software needs to be strong and industrial quality to maintain. For this Price may will be a bit high but in the long run we will be able to give best quality insurance.

**3. Functional & Nonfunctional Requirements:**

**3.1 Non-Functional requirement:**

1. Providing doctors with accurate results

2. Providing data for research purposes.

3. Suggesting proper medicine and diet for a certain disease.

**3.2 Functional requirement:**

1. Finding the disease by cross checking inputs (symptoms).

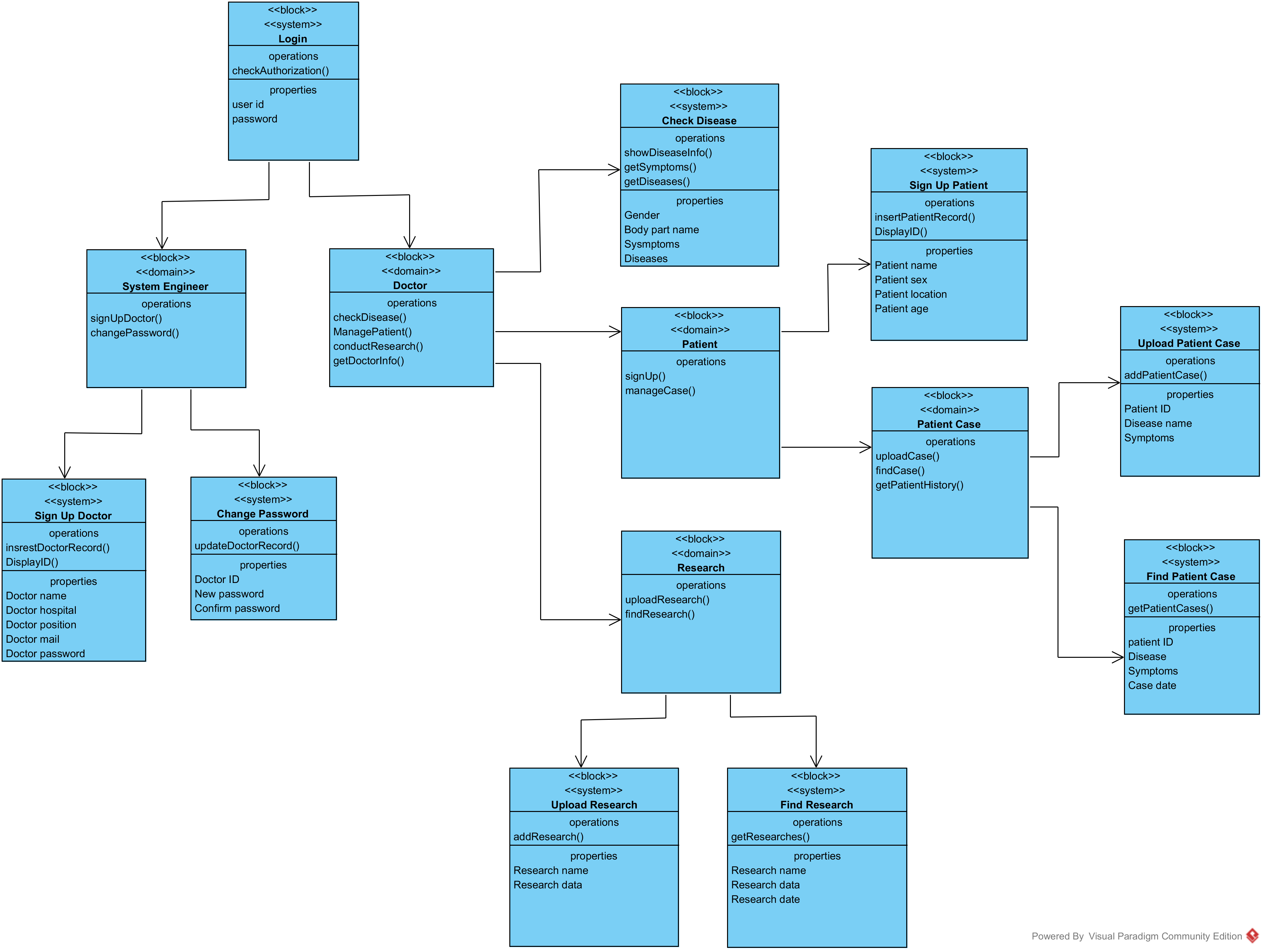
2. Generating symptom suggestions

3. Searching for medicine and diet suggestions from the database for the found disease.

**4. Design Procedure:**

**4.1 Architectural design:**

**4.1.1 Block Diagram:**



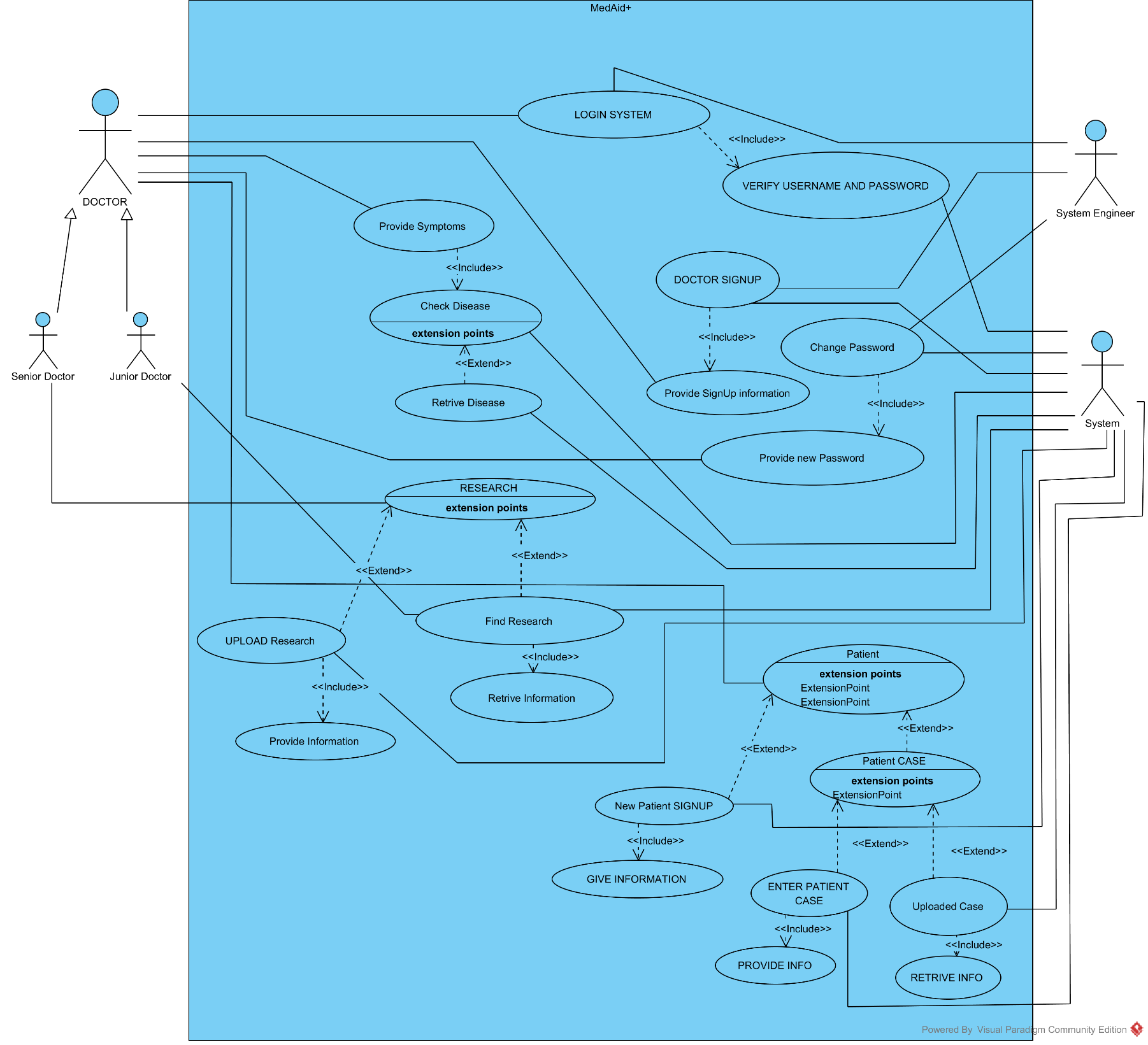
**4.1.2 Block Diagram Description:**

Login page redirects to either system engineer frame or doctor frame. Doctor frame can be redirect to symptom check frame or manage patient frame or manage research frame. Senior doctor can have access to any of this while junior doctor can not upload research.

Manage patient has 3 features- Signup new patient, upload patients case and view patient case. Manage research frame have the features of upload research or

Find research. System engineer can do 2 part- sign up new doctor and change existing password of a doctor.

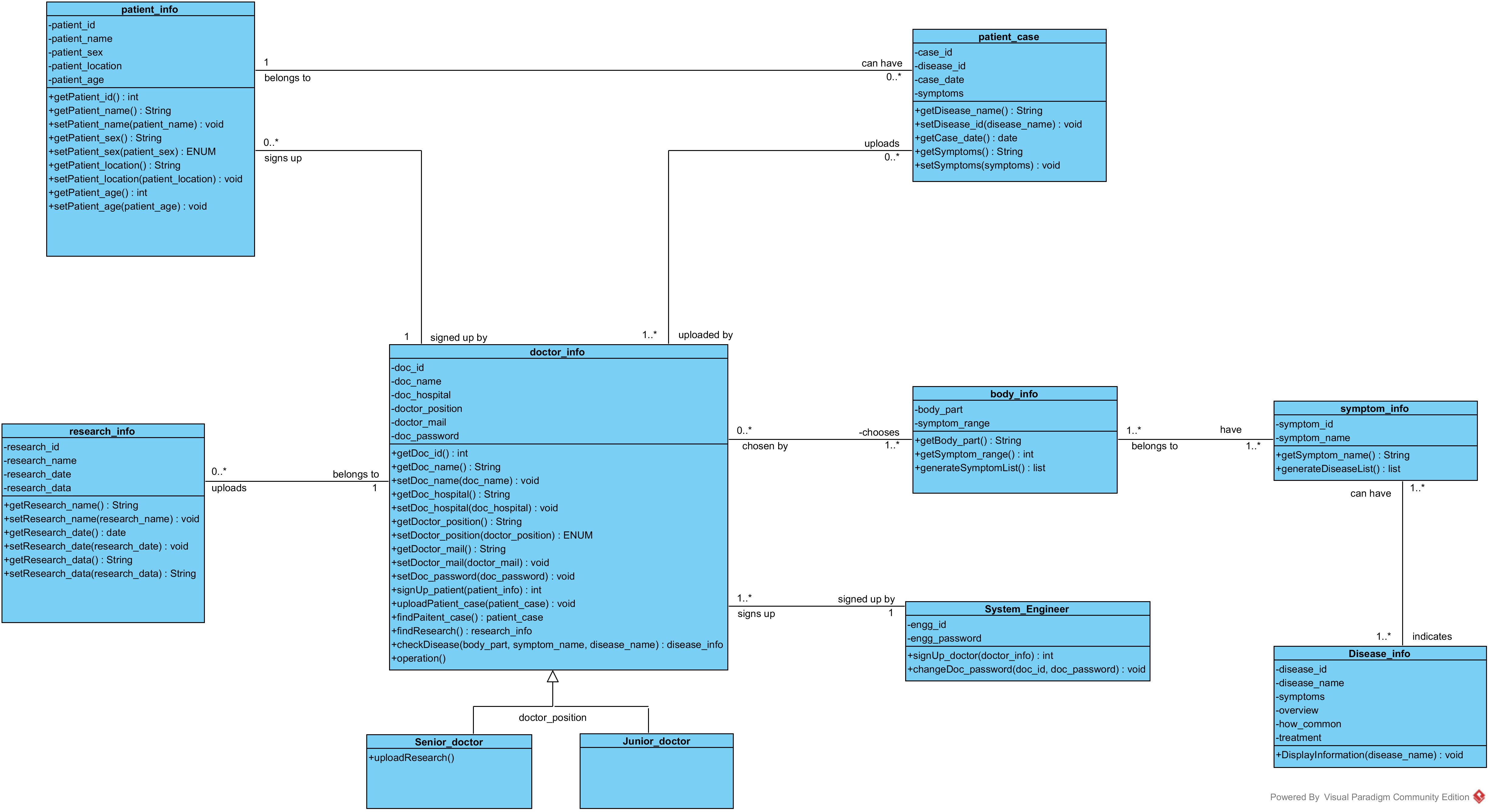
**4.1.3 Use-Case Diagram:**

****

**4.1.4 Use Case Description:**

|  |  |
| --- | --- |
| ID: | 01 |
| Title: | MedAid+ -A Doctor Helping tool |
| Description: | This Use case is to Assist doctors in making accurate decision makings. And Collect and provide data for research purpose to predict epidemics. |
| Primary Actor: | Doctor-(Senior Doctor, Junior Doctor) (generalization), System Engineer, System(database) |
| Preconditions: | * A valid User (any Doctor or least a System engineer) * Some Simple Symptoms Input from WebMD to start initially. |
| Post conditions: | * Doctors gets help from the software about the symptom patient has * Doctors will be able to see patient Case * Senior Doctors will be able to Update and see Uploaded research * System engineer will be able to make changes to any doctor’s password |
| Main  Success Scenario: | * **Doctors** should be **generalized** with **Senior Doctor** and **junior Doctor** as **actor.** * **Senior Doctors** can Do **Research** where he can do Two things **Upload Research** or **Find Research** * **Junior Doctors** will have only **Access to Find Research** * **Upload Research** will be done by giving proper information must and **Find Research** will be done by **Finding Research** and must retrieve data from database. * **Patient case** can be accessed by **Both Doctors** where they can do **New Patient Signup** or **Check patient Case.** * **Patient Signup** must have to give proper patient Information. * **Patient Case** have two option either can **upload patient case** or **can check patient disease** from database for both doctor must have to give proper input. Patient Case can be exported to pdf also doctors all uploaded case can be exported to pdf. * **Symptom checker** is accessed by **both doctors** and by giving proper input system must give the possible disease. * **System engineer** can **Create new Doctor** or He can **change doctors password** and both have to give proper information * **Patient** and **Doctor** both must have to have **Unique ID** on **signup** and will be needed for query |
| Extensions: | * To login it has to give proper USERID and PASSWORD or it shouldn’t login * To find Doctors Uploaded case USER have to give proper ID otherwise it shouldn’t GIVE output * To find Patient Case to PDF doctor has to give proper patient ID * System only should show proper diseases for proper symptoms. |
| Frequency of Use: | Often by Many Doctors mostly |
| Status: | Prototype Done |
| Owner: | RMS (CSE470)Group |

**4.1.5 Class Diagram:**



**4.1.6 Class Diagram Description:**

A dotcor can sign up new patients and upload their cases where which case

belongs to which patient and uploaded by which doctor is saved in doc\_patient

table. The primary key here is the concatenation of doc\_id, patient\_id and case\_id.

This table mainly serve as an intermediary table to break down many to many

relationship.

In patient\_info table, we are keeping patient ID as primary key and other

informations and in patient\_case table, we are keeping the case\_id attribute as

primary key.

A patient case will indicate a certain disease that the patient was affected by and

the disease\_id is a foriegn key here pointing to the primary key disease \_id of

disease\_info table.

disease\_info table also keeps symptoms for a disease which points to the

symptom\_info table through disease\_symptom table for breaking down many to

many relationship. This tables primary key is joint of foreign keys,

disease\_cross\_id and symptom\_cross\_id referring to disease\_info and

symptom\_info table. symptom\_id is the primary key of symptom\_info table.

As a body part can have many symptoms, we have one to many relationship from

body\_info to symptom\_info table. body\_part attribute is unique and primary key

here.

As a doctor can upload many researches and a research data can be pulled by many

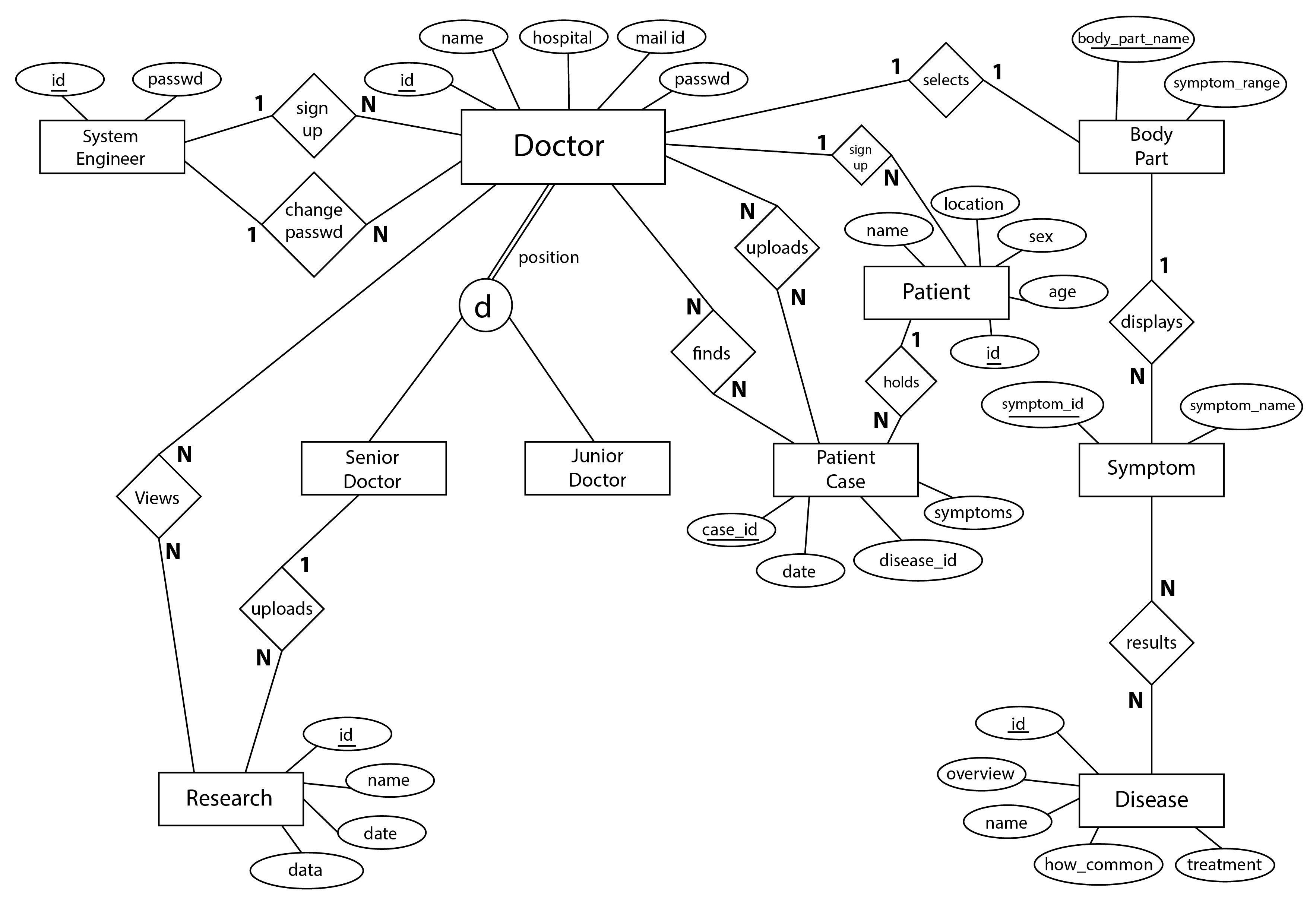
doctors, doc\_research is the table holding doctor\_id and research\_id together

primary key. research\_id here is a foriegn key pointing research\_id attribute of

research\_info table.

**4.2 Database Design:**

**4.2.1 Extended ER Diagram:**



A dotcor can sign up new patients and upload their cases where which case

belongs to which patient and uploaded by which doctor is saved in doc\_patient

table. The primary key here is the concatenation of doc\_id, patient\_id and case\_id.

This table mainly serve as an intermediary table to break down many to many

relationship.

In patient\_info table, we are keeping patient ID as primary key and other

informations and in patient\_case table, we are keeping the case\_id attribute as

primary key.

A patient case will indicate a certain disease that the patient was affected by and

the disease\_id is a foriegn key here pointing to the primary key disease \_id of

disease\_info table.

disease\_info table also keeps symptoms for a disease which points to the

symptom\_info table through disease\_symptom table for breaking down many to

many relationship. This tables primary key is joint of foreign keys,

disease\_cross\_id and symptom\_cross\_id referring to disease\_info and

symptom\_info table. symptom\_id is the primary key of symptom\_info table.

As a body part can have many symptoms, we have one to many relationship from

body\_info to symptom\_info table. body\_part attribute is unique and primary key

here.

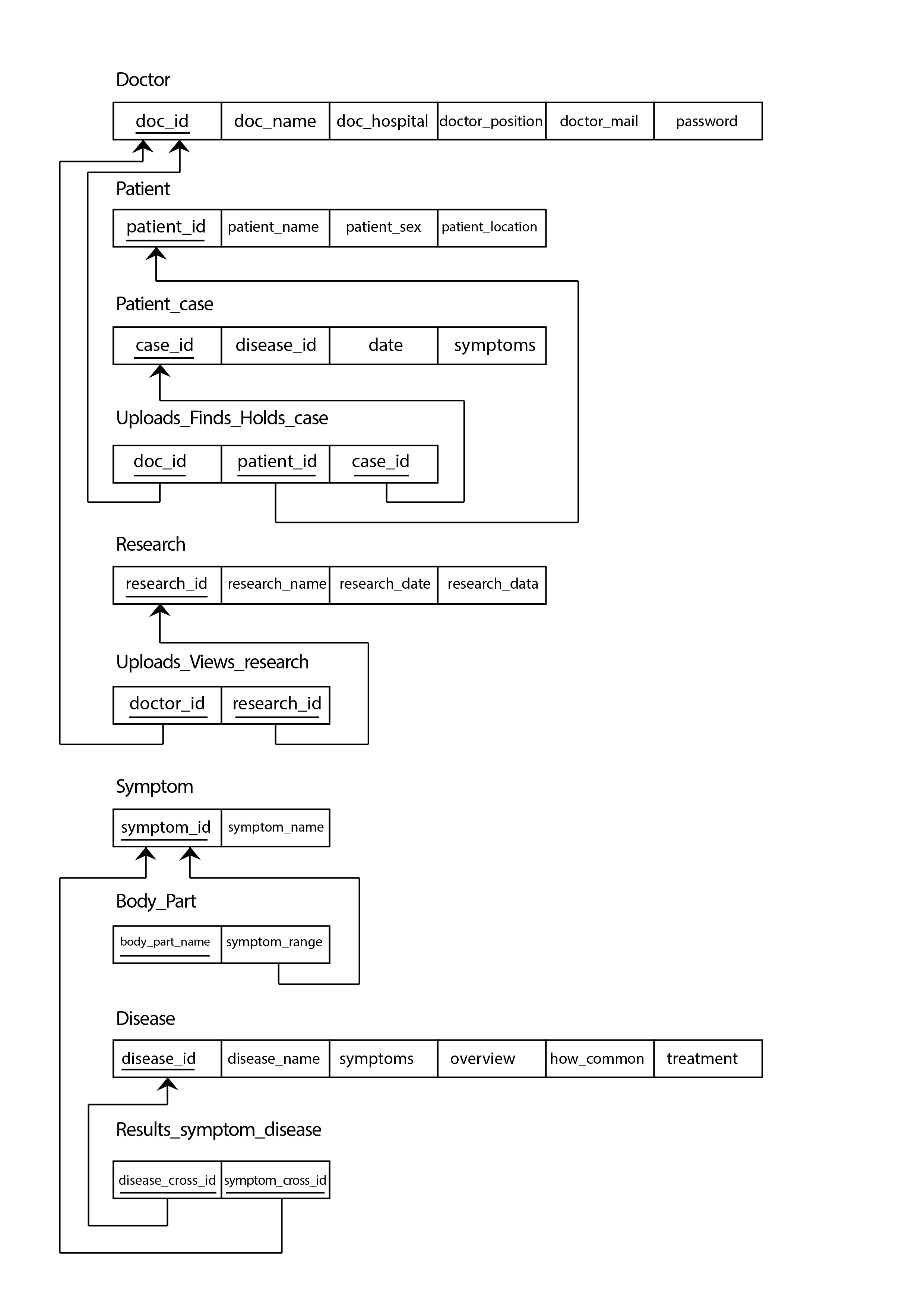
As a doctor can upload many researches and a research data can be pulled by many

doctors, doc\_research is the table holding doctor\_id and research\_id together

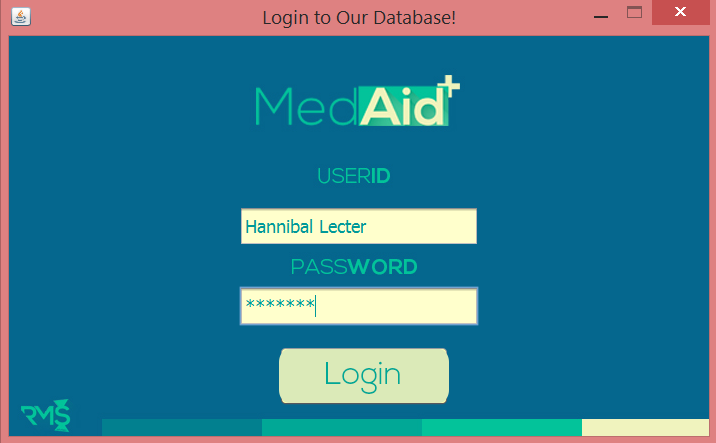
primary key. research\_id here is a foriegn key pointing research\_id attribute of

research\_info table.

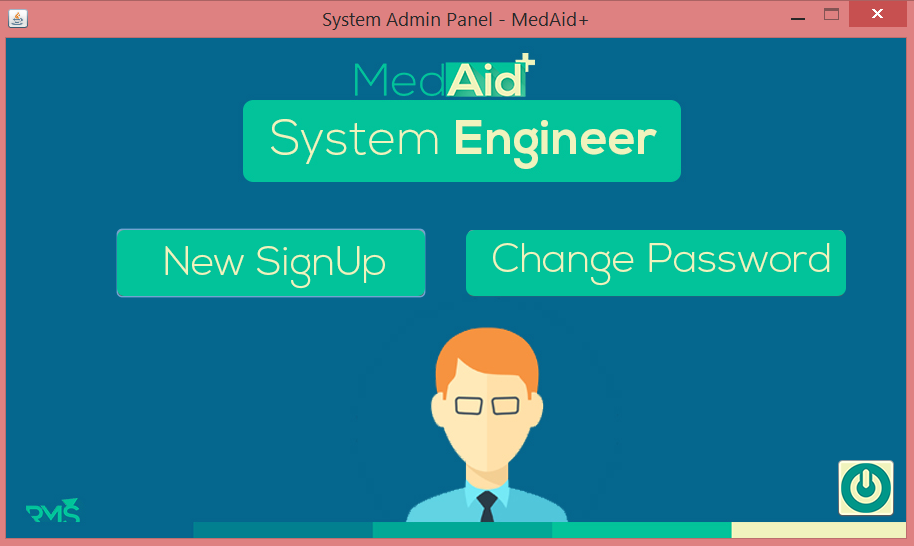
**4.2.2 Schema Diagram:**

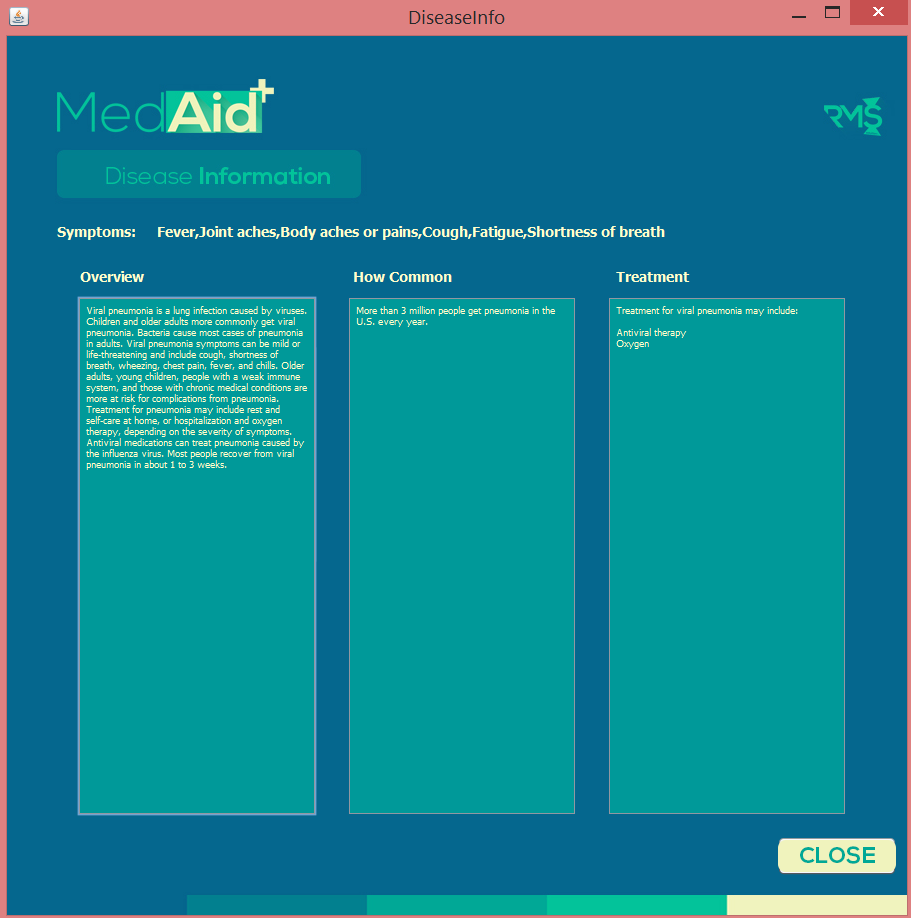


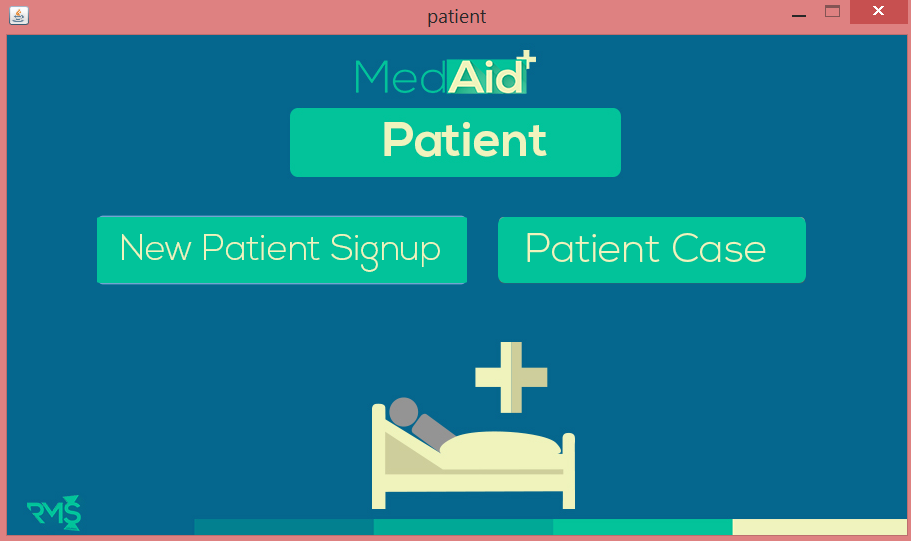
**4.3 User Interface:**













**4.4 Tool Used:**

We used,

1) Visual Paradigm

2) Java Swing

3) NetBeans

**5. Testing And Planning:**

**5.1.1 Test Plan Overview:**

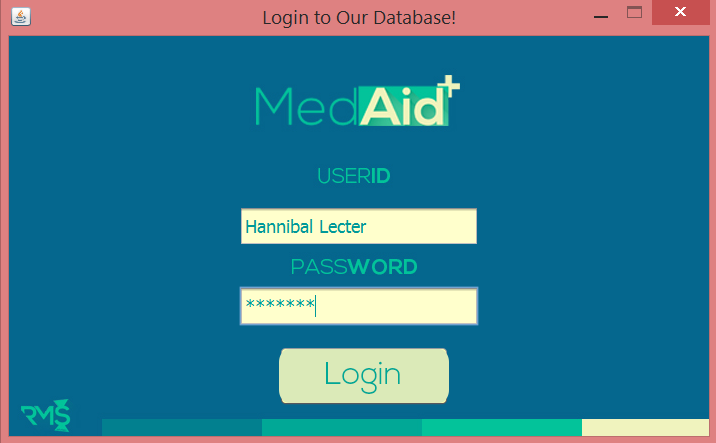
As MedAid+ is software intended for doctors to use, ensuring that it provides accurate information to the doctor is a must. In order to make sure MedAid+ provide correct information we have tested it to make sure there is no margin of error. Target of this test plan is to verify whether all the functionalities of MedAid+ is working properly and giving desired output for given sets of inputs. We have approached unit testing method to test each module of our software. There various functionalities that will be tested which are listed below:

1. Login Testing
2. Symptom checker functionality
3. Testing Disease Inf
4. Testing New Patient Sign-up module
5. Testing Enter New Case function
6. Testing Patient Case
7. Testing Upload Research
8. Testing Find Research
9. Navigation throughout the software is tested, testing whether each and every frame opens or closes properly

**5.2 Unit Testing Modules or Functions:**

1. Login Test, testing whether login system of MedAid+ is taking user name and password, and logs the user to its designated user panel or frame.

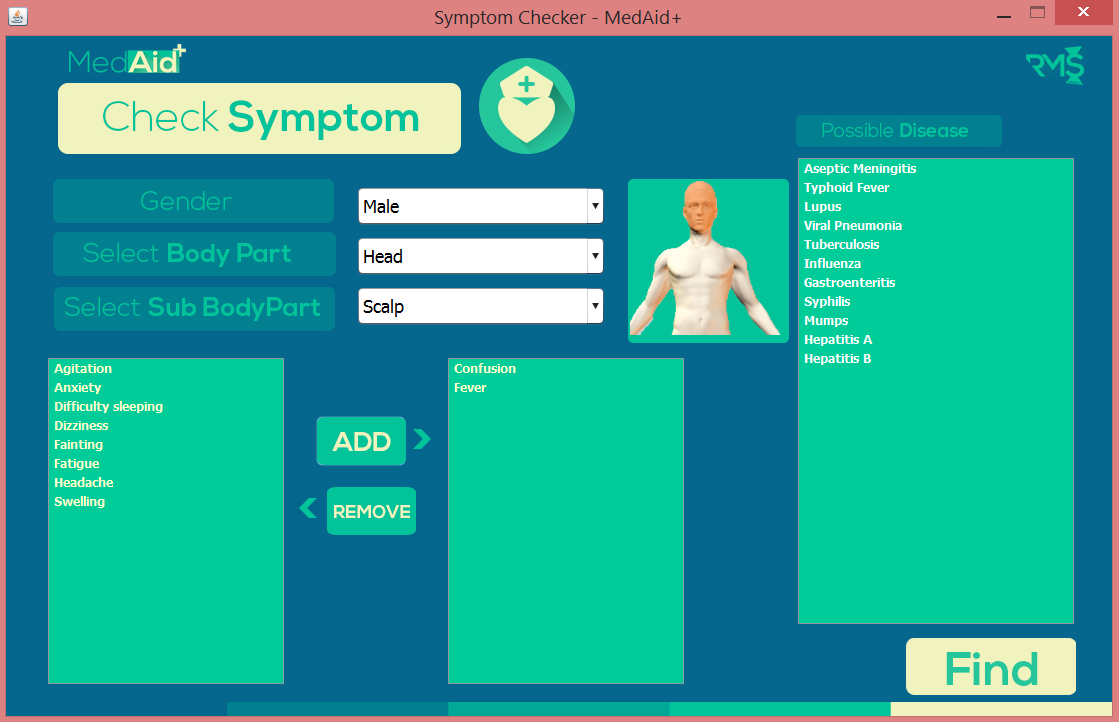
|  |  |  |  |
| --- | --- | --- | --- |
| Input ID | Input password | Expected response | Match |
| 1 | yodrake | Senior doc login | Yes |
| 4 | human555 | Junior doc login | Yes |
| sys | -empty field- | System engr. Login | Yes |
| 9 | card | Wrong pass or username | Yes |



1. Symptom checker functionality, testing whether selected body parts displays possible symptoms in the symptom box also whether pic of body part changes with diff body part, testing whether symptom checker gives accurate diseases output for particular set of symptoms

|  |  |  |  |
| --- | --- | --- | --- |
| Selected body part | Selected symptoms | Expected diseases | Match |
| Head – Scalp | Confusion, Fever | * Aseptic Meningitis * Typhoid Fever * Lupus * Viral Pneumonia * Tuberculosis * Influenza * Gastroenteritis * Syphilis * Mumps * Hepatitis A * Hepatitis B | Yes |
| Head – Scalp | Headache, Swelling | * Acute Sinusitis * Aseptic Meningitis * Typhoid Fever * Brain Tumor * Migraine Headache * Lupus * Mumps | Yes |





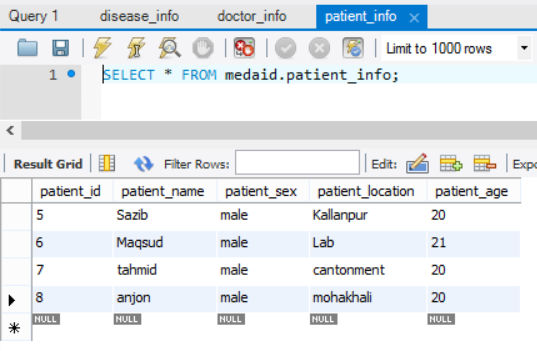
1. Testing Disease Info, testing whether selected disease information such as symptoms, overview, and treatment are accurately shown in display.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Selected Disease | Expected Overview | Expected  How common? | Expected Treatment | Match |
| Aseptic Meningitis | Aseptic meningitis, or viral meningitis, is an inflammation of the lining around the brain and spinal cord. It's usually caused by a virus. Less often, fungi, diseases spread by insects, and other conditions trigger it. Aseptic meningitis is usually not as serious as bacterial meningitis, which can be life-threatening. Symptoms include fever, headaches, neck pain and stiffness, nausea, fatigue, confusion, and sensitivity to light. Most often, aseptic meningitis goes away on its own with rest, fluids, and pain relievers. However, any signs of meningitis are always a medical emergency. The symptoms of aseptic meningitis are often the same as those of bacterial meningitis, which is life-threatening without treatment. | About 1 in 9,000 adults and children develop aseptic meningitis each year in the U.S. | Treatment for aseptic meningitis caused by a virus involves getting plenty of rest and taking over-the-counter medication for fever, headache, and muscle aches. Antibiotics won't cure viral meningitis. If something other than a virus caused the illness | Yes |
| Influenza | The flu is a respiratory tract infection caused by the influenza virus. Flu symptoms include fever, sore throat, runny nose, headache, cough, muscle aches, and fatigue. The flu spreads easily from person to person. Most people recover at home with rest and over-the-counter medications, but for some people flu can be severe and even life-threatening. Young children, older adults, pregnant women, and people with health problems are at risk for complications from the flu, such as pneumonia. Prescription antiviral medication can lessen symptoms and help speed recovery. Getting a yearly flu vaccine is the best way to prevent the flu. | In the U.S., about 5%-20% of people get the flu every year. | The flu usually gets better on its own in one to two weeks with rest. Antihistamines, decongestants, pain relievers, drinking plenty of fluids, and inhaling steam may help ease symptoms. For people at risk for complications and those with severe symptoms, antiviral medications such as oseltamivir (Tamiflu) or zanamivir (Relenza) may help lessen symptoms, shorten the illness, and prevent complications if they are taken soon after symptoms appear. | Yes |

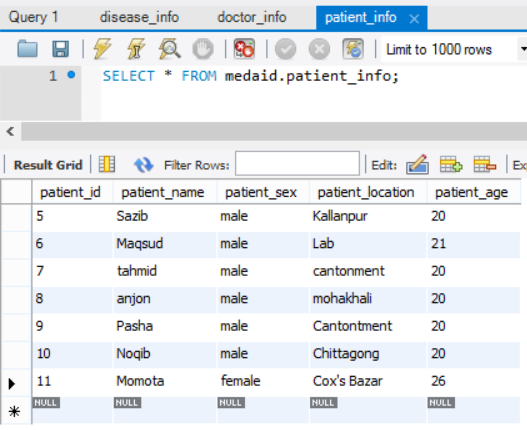
1. Testing New Patient Sign-up module, testing whether new patient sign takes data and saves them in the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Patient Name Input | Age Input | Sex | Location Input | Database status |
| Pasha | 20 | male | Cantonment | Update success |
| Noqib | 20 | male | Chittagong | Update success |
| Momota | 26 | female | Cox’s Bazar | Update success |

Initial Database:



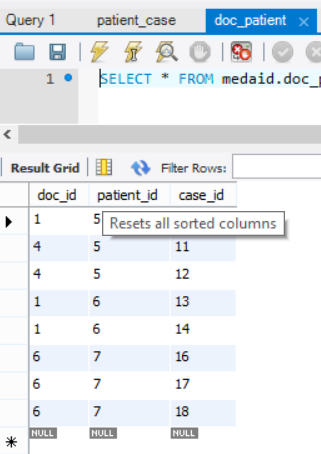
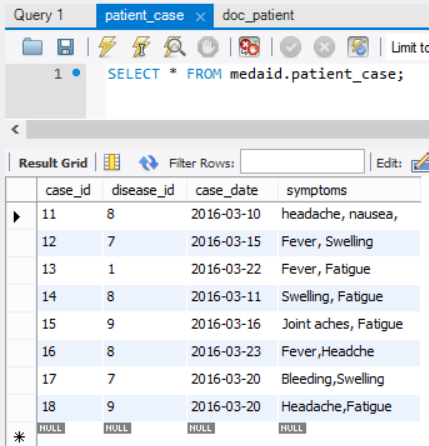
Database after Input:



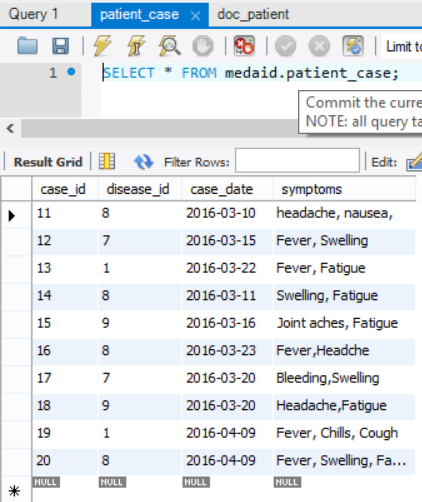
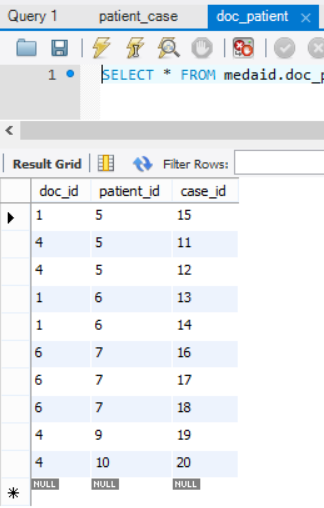
1. Testing Enter New Case function, checking whether patient’s disease and symptoms is saved in against the patient ID in the database along with date of upload.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Patient ID | Input Disease | Input Symptoms | Database status |
| 9 | Influenza | Fever, Chills, Cough | Update success |
| 10 | Mumps | Fever, Swelling, Fatigue | Update success |

Database before input:



Database after given input:

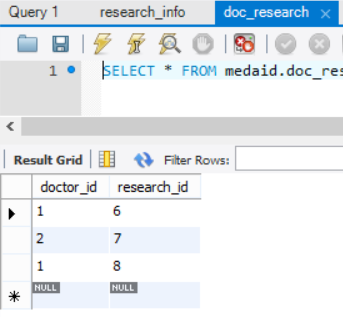
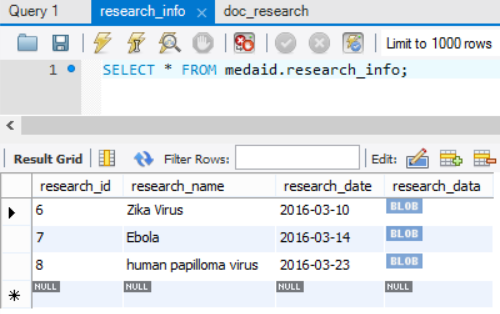
1. Testing Patient Case, checking whether patient info such as disease, symptoms and date of upload can be retrieved by patient ID correctly.

|  |  |  |  |
| --- | --- | --- | --- |
| Patient ID Input | Diseases retrieved | Symptoms Retrieved | Match |
| 9 | Influenza | Fever, Chills, Cough | Yes |
| 10 | Mumps | Fever, Swelling, Fatigue | Yes |

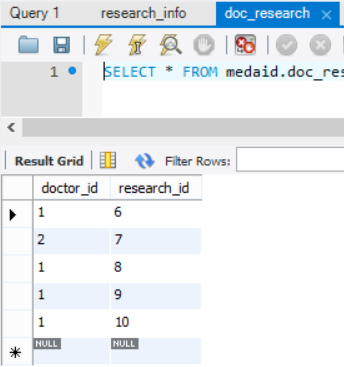
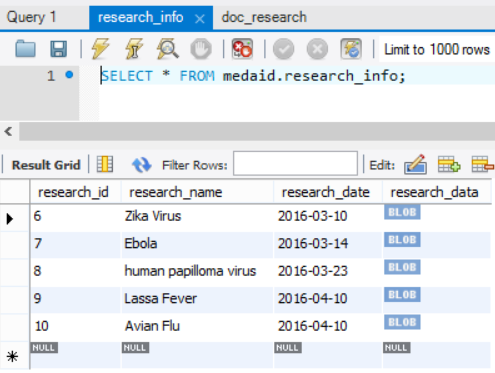
1. Testing Upload Research, testing out whether research name and data is successfully stored in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Doc ID | Research Name Input | Data Input | Database status |
| 1 | Lassa Fever | … lassa fever research data…. | Update success |
| 1 | Avian Flu | … avian flu research data …. | Update success |

Database before given inputs:



Database after given inputs:



1. Testing Find Research, testing whether all research that have been uploaded are available in the drop down list, and selected research name shows data related with that research.

|  |  |  |
| --- | --- | --- |
| Research selected | Data retrieved | Match |
| Lassa Fever | .. lassa fever research data ... | Yes |
| Avian Flu | .. avian flu research data … | Yes |

**5.2 Performance Analysis:**

While carrying out the above test MedAid+ did not have any issue with speed correctness and security. The performance was upto the mark in testing phase, however as time will pass the patient case data and research stored in the database will increase in size, this may affect the data retrieval time in future.

**6. Conclusions:**

**6.1 Summary:**

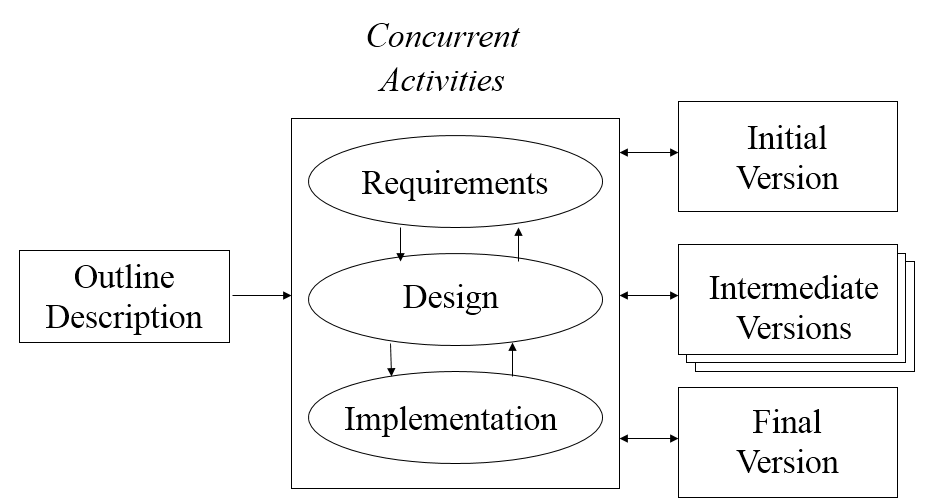
There are many sectors where computer science is doing huge advancement such as military, Agriculture, Medical sector etc. But in Bangladesh Medical sector don’t have that much advancement on computer science still some doctor uses analog system to check their patient this is exactly why we choose to work on medical sector and do help this kind of issues.

Our system will provide three main features which has sub features parts. Now System will have three view senior doctor, junior doctor and System Engineer where System engineer will help with doctor login information or signup but senior doctor can upload Research and Check all record from the system. Where junior doctor will have only find research access. Junior doctor won’t be able to upload research because he is not yet experience to do this. Other two main features are one Symptom checker where both doctors will be able to check symptoms providing epidemics & it will also show how common the disease is, how to stop it etc. Last but not least is patient Record system, where patient will be able to sign up by doctors and doctors can upload patient case and also see them using patient ID. In this system user (doctor) have to remember his User ID which generated on signup to login in the system. In patient record system doctor can export all record of patient to PDF. In addition, Doctor can also export how many case a doctor uploaded to see his previous data upload. This way doctors can check which doctor is doing research or uploading patient case on what. In symptom checker result will be more accurate over the time so value of this software will increase.

Therefore we can see a whole system overview here, this product will bring huge change to medical sector. Senior doctors who are doing research will get huge benefit from it and junior doctors will get huge benefit from symptom checker because it will help them to get more accurate result and will give them better experience so that when they will be senior they will be able to provide much better doctors in our society.

**6.2 Discussions on methodologies and potential impacts:**

We followed Evolutionary development. Evolutionary development have started on initial version. We had to test the software so after implementing tested the software and by getting result of it we make intermediate version and then after adding specific features completely it will be our final version so version control was one of the major thing in our methodologies. We didn’t go for waterfall model because doctor might need new requirement so re designing would be huge problem. We used version make a symptom checker then another version has patient record system with symptom checker other version research function with both other two features. This way we were able to reach our final version.



**FIG 6.2.1: Evolutionary Development**

Potential impact will be high in the long run. In short run, our system will have less data on our database therefore it will have less impact then long run, but even if as we are using one database (WebMD) as our initial database therefore it will show results close to symptoms but in long run it will be more accurate. Furthermore, as our User Interface is interactive this will impact on end user as it is pretty easy to use.

To conclude, we can say that our software will have huge impact in long run. Initially impact will also be good as our Business risk is less.

**6.3 Limitations & Future Works:**

**6.3.1 Identify System Limitations:**

The Software Limitations are,

* Doctors need to be trained to use the system properly
* Doctors need to insert new information correctly otherwise result will be affected by incorrect data.
* As the number of servers are limited, if one server crashes it will largely affect the other servers.
* As we are using one database now, for the long run we might need to add more database in the system.
* This system is limited to personal computer.

**6.3.2 Future Works:**

Future Possible Works are,

* Implement a AI which can check the case and remove less important case
* Chat system over logged in Doctors who can chat among themselves
* Introduce as website and android app
* Making dashboard where doctor can ask question in software and other doctors can reply

MedAid+’s possibilities are countless, Even we can take doctors feedback through software to update they whole software which has possibility to bring a new era to medical sector.