**Contents**

1.INTRODUCTION

1.1 Purpose…………………………………………………………………..... 2

1.2 Scope……………………………………………………………………… 3

1.2.1 Use Case Model…………………………………………………… 4

1.3 Definitions,Acronyms and Abbreviations………………………………. 5

1.4 Glossary…………………………………………………………………… 5

1.5 Overview………………………………………………………………….

2. Overall Description

2.1 Product perspective ………………………………………………………. 7

2.1.1 Business Case, Operational Concept, Current Solution………… 8

2.1.2 System, Software and Hardware specification…………………. 9

2.1.2.1 System and software specification………………………. 9

2.1.2.2 Hardware specification…………………………………… 10

2.1.2.3 Memory,Operational features and communication……. 10

2.2 Product Functions……………………………………………………… 11

2.2.1 Supported Functions……………………………………………….. 12

2.2.2 Unsupported Functions……………………………………………. 12

2.3 User Profiles…………………………………………………………… 13

2.4 Constraints ………………………………………………………………. 13

2.5 Assumption and Dependencies……………………………………….. 13

3. Specific Requirement

3.1 Functionality………………………………………………………………….. 14

3.1.1 Main Features…………………………………………………………. 16

3.1.2 Maintenance Functionality………………………………………….. 16

3.1.3 Graphical User Interface……………………………………………. 17

3.1.4 Interfaces to external systems……………………………………… 17

3.2 Usability

3.2.1 Language………………………………………………………………. 18

3.2.2 History…………………………………………………………………… 18

3.3 Reliability

3.3.1 Availability…………………………………………………………….. 19

3.3.2 Error Rate……………………………………………………………….19

3.3.3 Error Handling………………………………………………………… 19

3.4 Security ………………………………………………………………………. 20

3.5 Hardware…………………………………………………………………….. 20

3.6 Performance………………………………………………………………. 20

3.7 Deployment………………………………………………………………… 20

3.8 Operating system…………………………………………………………. 20

3.9 Legal Copyright, licensing and online user documents ………………. 20

3.10 Interfaces, communication between devices……………………….. 21

4. Change Management Process…………………………………………………… 21

5. Document Approvals……………………………………………………………… 21

6. Supporting Informations…………………………………………………………. 21

7. Reference………………………………………………………………………….. 21

8. Tools………………………………………………………………………………….. 21

**1.Introduction**

**1.1 Purpose**

This document specifies the software requirements specification (SRS) for the “**Visualization of Genetic Algorithm With Maxone Problem”.** It describes the scope of the system, the usability and application of the system both functional and nonfunctional requirements for the software, the design constraints and specification of the system and system interfaces. The name of our software we have given as **GENOME 1.0.**

**1.2 Scope**

Our project **GENOME 1.0** is basically an educational project. It actually visualizes the activity of or in a word the application of “**Genetic Algorithm”.**

“Genetic algorithm” is actually a probabilistic and heuristic algorithm. It’s actually an optimization algorithm. The main theme of genetic algorithm lies on these following basic theme - “***How the best Gene is selected for the next generation. How can we deside and determine the characteristics of the next generations. How can we choose such a combination and merge them so that we can derive the best possible future generation.”***

So from these basic questions, our algorithm comes and gives us a way to decide the proper way to get the best outcome. And for this reason our algorithm is basically an optimization algorithm. And for this optimization characteristics, we can use this algorithmic idea to many fields in this present world to solve the practical problems that we are facing daily. And this algorithm will give us scope to get the best outcome.

There are many application of this supreme algorithm. But basically here, we are interested in

it’s one of the most powerful application **“MAX ONE PROBLEM”.** Our goal is to visualize this algorithm with this application.

This project will give us a platform where we can make people know about the importance of this algorithm, show how the total basic and design is standing. This document will include the **functional and nonfunctional requirements, use case diagram, the visualization model architecture, the system functionality and workflow design and algorithmic implementation** of our software **GENOME 1.1.**

**1.2.1 Use Case Diagram**

To define the characteristics, application, and functional requirements of our project here we will describe use case diagram. It will define functionality of our software, along with the application using steps , the workflow of using this software along with the interfacing relation between the users and shell and core of our program. The use case model consists of the user and software interaction and the functional behaviour of our program.

**1.3 Definitions,Acronyms and Abbreviations**

| Names | Abbreviations |
| --- | --- |
| Rid | Request Id to determine the name of functional requirement. |
| T | Title .to represent the subclass of the request. |
| P | Priority, basically to determine the importance of the procedure. |
| R | Risk, which will consider all other respective fields of some step from the procedure. |
| IEEE | Institute of Electrical and Electronics Engineers, which defines various standard on many respective fields of science. |
| User\_i | User\_i basically represents various level of users. Where i defines different parameters. |
| API | Application program interface |
| GUI | Graphical User Interface |
| Swing,fx | Java GUI libraries |
| Awt | Java applet and package for designing. |

**1.4 Glossary**

| Name | **Description** |
| --- | --- |
| Genetic Algorithm (GE) | Probabilistic and heuristic algorithm to choose the best possible outcome from a certain pattern of gene combination. |
| Max One Problem | Application of genetic algorithm comprising of only two digits zero and one. |
| User choice /Random choice | There are basically two level of choice in the program. One is user level choice and other is random choice which precisely defines what will be visualized if user gives choice and if random choice is performed. |
| Customization | Generally an option to set the environment on what basis the visualization will be performed. |
| Authorization | The level through which the user will lend or change the authority from other subordinate account users. |
| Interface | The medium through which two levels are connected. |
| Table 2.1 | Describes the users of the procedure. |
| Table 2.2 | Describe the depth workflow |

**1.5 Overview**

Chapter 2 defines the general product functions, characteristics, application, process logic and speciality. Chapter defines the intended applications,constraints to be respected and the assumption made in order to define the requirements.

Chapter 3 mainly describes the functional and nonfunctional requirement usability,reliability,security,performance and maintainability considerations and requirements to a level of detail sufficient to enable designers to design, to the developers to develop to make the application to a certain level and the testers to test that the system satisfies these requirements.

Chapter 4 contains index, appendices and other supporting information and small details.

This SRS document is prepared by following IEEE 830 - 1998 Standard.

**2.Overall Description**

**2.1 Product Perspective**

**2.1.1 Business Case and Operational Concept, Current Solution**

Our present solution is nothing but just simple implementation of the precise genetic algorithm using java graphics libraries, xml and using intelligent IDE , android studio.

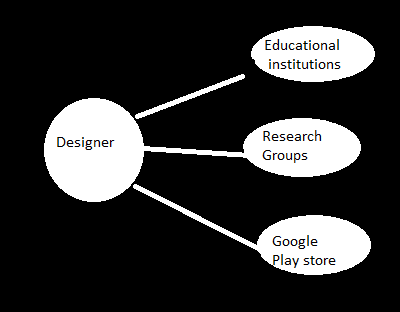
**Genome 1.1** is basically a software for educational purpose. In general life, this software does not have many issues except giving people the ideas about how genetic algorithm is used in our life to find the best possible outcome. We actually visualize the application of this algorithm.

This will be done very attractively and it will be an android operating system based software. Here graphical interface will be used very beautifully such that it can attract people to use this software and to understand the core concept of the precise algorithm.

As it’s an educational software, it has been designed as very user friendly, along with many user needed options so that they can manipulate or change the interface so that they can understand the way they want to. We are actually giving them a platform to understand and so this should be open.

So, actually our software can not be approached to general people. But the most important part, these can be approached to that group of community who are actually connected with the purpose of teaching people. Mainly, Educational institutions, research institutions and large companies who have the job of either teaching people or have the importance of learning this algorithm and to understand the deepest application of it. We will actually approach our product to them. Besides, it will be up in Google Play store. So, other interested people can easily get access to it if they are interested.

So in General our business Diagram we can actually see this scopes as marketing:



**2.1.2 System, Interfaces, software and Hardware Specification.**

**2.1.2.1 System and Software Specifiction**

Our whole software is actually based on **Android operating system**(AOS). It’s one of the most usable operating system in almost all of the smart electronic devices nowadays. And the most important part almost now a days it is the most usable, convenient and easily accessible operating system now. And people are at a huge amount using it. **Ice Cream Sandwich 4.0** is at least needed to run our software and **minimum requirement of needed API is 14**.

**2.1.2.2**  **Interface Design**

We will use precise GUI (Graphical User Interface) to draw the attention of the user. For these purpose we are using **JAVA swing, Javafx, Java awt packages**. Besides we are using **XML** to give it a dynamic version. Besides we are using **android Studio (25)** as our IDE . Besides this is actually an example of our proposed idea.

We are planning to make such interface for our design. Besides here we want to as much swiftness as we can. These features will attract the consumers and we are hoping these characteristics will create much enthusiasm among the viewers and the users. And will fulfill their thirst of learning and will encourage more and more to use this software.

**2.1.2.3 Memory,operational and communication**

Our software do not need any additional hardware requirements. It’s totally software depended and it does not have any extra hardware dependency. Normally any device which has **java development kit (JDK)** can successfully run an android operating system, it’s fine, sufficient and enough for our system. Our program can successfully run there.

Normally extra memory or storage device is not needed. It’s just enough to have any **SD card** of 2.0 GB in that device. We have also designed a web page and facebook page to get the opinion and demand from our software users. As world changes new ideas and modification comes. So, from their review we can develop the characteristics of our product.

‘Our precised facebook page - [https://www.facebook.com/Genome1](https://www.facebook.com/Genome1.1)

**2.2 Product Functions**

**2.2.1 Supported Functions**

Our **Genome 1.0** includes the following features:

1. Supports idealistic visualization of the famous genetic algorithm through max one problem.
2. Supports multiple mode of operation.
3. Supports multiple working environment on the basis of user’s choice from defined and various choices.
4. Provides interactive interface for the controlling of the software.
5. Supports Multiple operation at a time, multiple combination, merge and classification of the programs.
6. Gives opportunity to access the internet access to learn more about genetic algorithm, it’s core theoretical concepts also including applications.
7. Can capture the present visualization in memory on the basis of user’s command.
8. Do not need any extra hardware dependency.
9. Supports distributed and variable functionality.
10. Applicable for least to android to highest versions.
11. Supports swift, fast and attractive graphical visualization.
12. Provides theoretical conceptual part and breakdown of concepts and also can view them in through attractive graphical presentation.
13. Can have view hierarchy through different level
14. Supports both implementation of user input along with random input

And their visualization.

15)Can have account privacy systems and own input database management cluster

Including downloadable files and other necessary links and articles.

16) Applicable for any type of presentation with external other devices, Example - Powerpoint or video presentation along with computer and projector.

**2.2.2 Unsupported Functions**

1. Three Dimensional visualization can not be found.
2. Don’t have access level hierarchy
3. Don’t have bug tracking and auto change management
4. Artificial Intelligence level is not very high.
5. Environment Variability rate is not very build up

**2.3 User Profiles**

Our system is not actually depended on various users. We can divide our user class in following parts.

| User | Functions and Responsibilities |
| --- | --- |
| User\_1 | User will basically give his desired data set and our software will work on the basis of it, simulate it, visualize it and show it . |
| User\_2 | This type of user will actually ask software to simulate own desired pattern and demonstrate the output to other users. |
| User\_3 | Generally a combination of user\_1 and user\_2 who will use this to teach others. They are basically the leaders or teachers in educational institutions, research institute or in large companies. |

**Table - 2.1**

**2.4 Constraints**

This document is actually a study project, not a real life SRS. So there are many technical failings. It actually gives direction how to build up our project GENOME1.0 and also shows the necessary planning, requirements for the completion of the project. And at a whole gives a whole flavour of a real life software requirement specification. We have actually targeted a certain level of viewers (Researcher and Educational Institutions) mainly .We are giving higher priority in them and keeping option for the general viewers through google play store.

**2.5 Assumption and Dependencies**

Our present assumption is that we don’t need any extra hardware dependency at certain moment and the whole process is built up on this. But if in future, for development of the project we can have extra hardware,memory and graphics dependency. For this there can be some radical changes in our design and internal design architecture.

Besides, we are limiting our operating system only to Android and there can be time when we are thinking it to build as platform which is suitable for MAC,linux and windows operating system also. And as android is generally used on mobile application , basically our program is run in mobile devices or those devices which support android. We have a plan to implement in such a way that it is suitable for various types of electronic devices.

Besides, we are using java programming language and xml language for our development. Bit it would have been better such that we can use more multiple languages, libraries and bring better development approach.Our minimum version of android operating system is ice cream sandwich 4.0. So , we are guessing people have at least this version. But older then this will not give them scope to use or **GENOME1.0**

**2.6 Use Case Diagram**

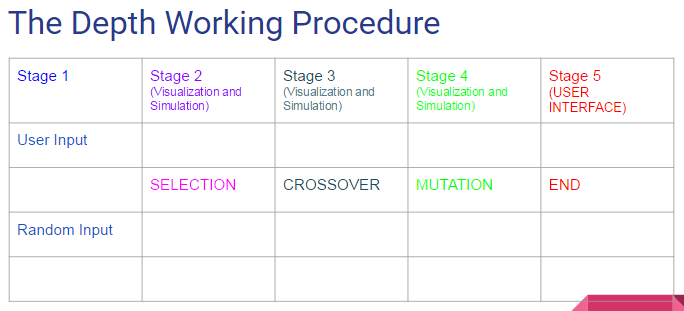
Use case model actually defines the users and their role in the system. It also defines the procedure hierarchy and the relations between the users. And beside that it also defines the accessibility and classification of work scheduling. It actually defines the functional requirements .Each use case describes how the system will be used by the actors to perform a specific goal.

Use case diagram do not capture non functional requirements of the system. Generally we only use some lines to define.the division and characteristics of the whole procedure. It just segments the total procedure and show the application classifications.

We have already explained the users of our system in **table 2.1.** Generally we have manager, project team leader, project team member, system administrator and general users.

We already defined in brief how the first four users will act. And now we actually want to define how ‘general users will perform. They will basically use the interface, through they will perform their part.

Generally we can show their part through the following table.



**Table - 2.2**

Our user, will give their input and from that we will visualize the steps. **SELECTION,CROSSOVER ,MUTATION.** Each of the steps will be visualized so that our audience can understand how we are choosing the genome pattern to find the best optimum solution on the basis of genetic algorithm and its applications. This represents the general work view of our customers.

**3.1 SPECIFIC REQUIREMENT**

This section contains all the functional and nonfunctional requirements. The functional requirements are grouped according using case model. A requirement has the following properties:

1)Requirement id(Rid): uniquely identifies the requirements within the application.

2)Title(T): Defines the functional group the requirement belongs to. Gives the

requirement a symbolic name.

3)Description(D): The definition of the requirement

4)Priority(P): Defines the order in which requirements should be implemented. Priorities are designated (highest to lowest) “1”, “2”, and “3”. Requirements of priority 1 must be implemented in the first system release.

5) Risks(R): Specifies risk of not implementing the requirement. It shows how the

particular requirement is critical to the system. There are following risk’s

levels and associated impact to the system if the requirement is not

implemented or implemented incorrectly .

• Critical (1) – the system will become unusable without meeting this requirement.

• High (2) – the system can be used minimally. But some of the significant features would lack. The main functionality would be affected.

• Medium (3) – some of the features would be hampered.

• Low (4) – the system can be used without limitation, but with some

workarounds.

**3.1 Requirements**

This section describes the main functional requirements of our project.

**3.1.1 Main Features**

**Requirement Id -** Genome/Rid/1

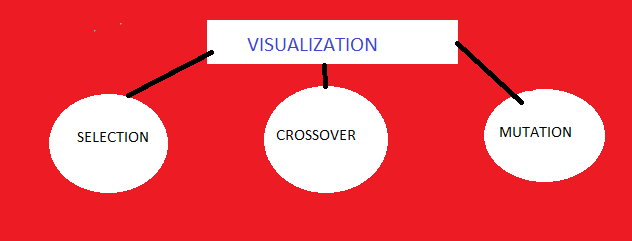
**Title -** Main Features/Visualization

**Description -** The visualization of genetic algorithm. Genetic algorithm is actually based on some basic operations. Selection, crossover, mutation. These basic operations defines the next level generations. Our main priority is to break the whole steps us to atomic level and show them graphically.

**Priority -** 1

**Risk -** 4

The whole procedure can be seen by following atomic levels



**Image 1 - Visualization atomic level**

**Requirement Id -** Genome/Rid/2

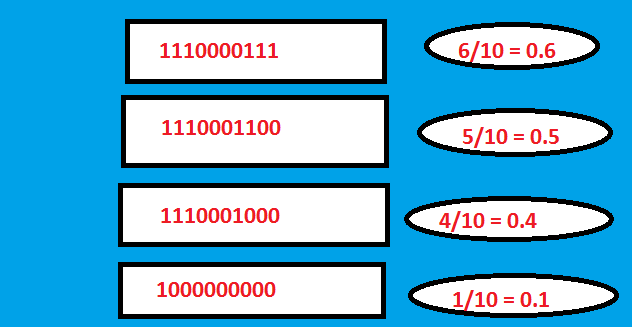
**Title** - Main Features/Selection

**Description** - Selection is the basic part of the three step parts of our gene evaluation. As we are discussing our algorithm on the base of Max One problem. This problem actually deals with the presence of binary bits . It says, the combination of ones and zeroes identically identifies new characteristics and statistically and probabilistically with excessive number of bits it proves the better gene. So our main goal is to visualize the chosen genes with the number of ones.

**Priority** - 1

**Risk**  - 4

The selection process is based on following algorithm. And we want to make this visualization in our program.



**Image 2 - Visualization Selection Level**

**Request id** -Genome/Rid/3

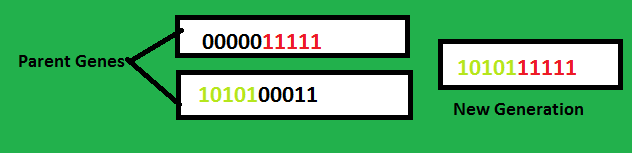
**Title** - Main Features/Crossover

**Description** - Crossover is the second stage of gene evaluation. So, there will be two choices. Either our users will ask to implement crossover on our opinion or they will choose a certain level of pattern and we will implement their desire.Maxone problem is actually a bit combining problem. So, the certain pattern is very necessary to identify this step.

**Priority** - 1

**Risk**  - 4

Genome will visualize something like this in crossover section.



**Image 3 - Visualization on Mutation level**

**Request Id** - Genome/Rid/4

**Title** - Main Features/Mutation

**Description** - The last stage of evaluation. We will show how after best selection and crossover stage we get a new mutated gene from the parent genes. This will be our third level visualization.

**Priority** - 2

**Risk**  - 4

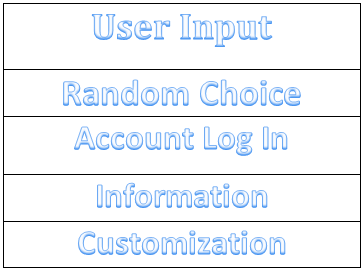
**Request Id** - Genome/Rid/4

**Title**  - Main Features/Choose options

**Description** - There will be options. The users can give their choices on which basis we will work. Or users can ask us to do what we desire. So, on that basis we will generate random inputs and then we will work.

**Priority** - 1

**Risk**  - 4



**Image 4 - User Interface**

**Request Id** - Genome/Rid/5

**Title**  - Genome/Multiple Running Application

**Description** - Our application is designed in such a way that at a same moment it can be run multiple times from the same device.

**Priority**  - 3

**Risk**  - 3

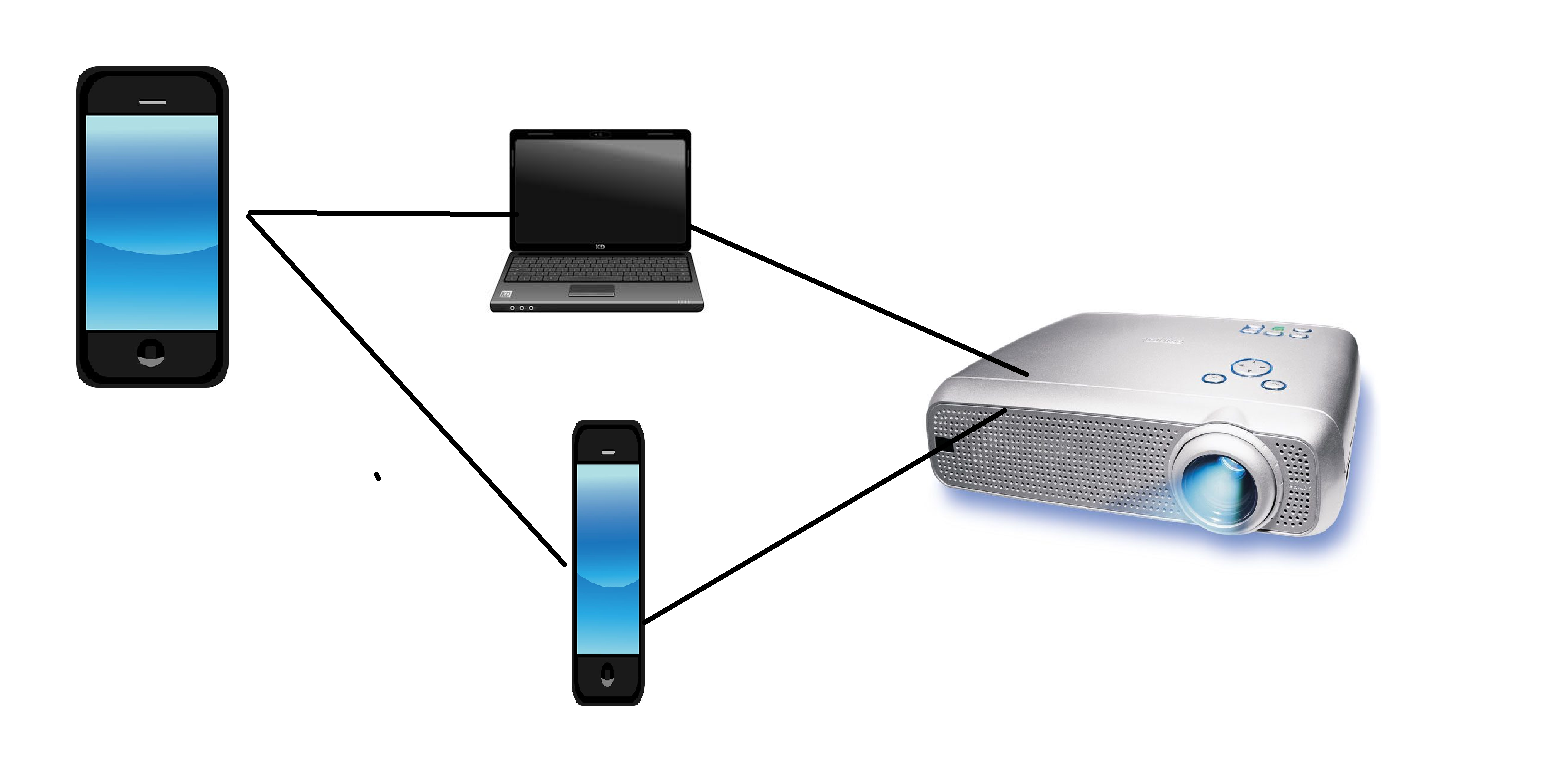
**Request Id** - Genome/Rid/5

**Title**  - Genome/Multiple Compatibility

**Description**  - Our application will be suitable for attaching with any other external devices such as projector, computer and other smart electronic devices for presentation, view or wide space visualization and in this regard we are actually building it so that it can be modified for any type of presentation purpose.

**Priority**  - 2

**Risk**  - 2

****

**Image 5 - Multiple Compatibility**

**3.1.2 Maintenance Functionality**

**Request id** - Genome/Rid/5

**Title** - Maintenance Functionality/Data Maintenance

**Description** - Our application is connected with the central database application of genetic algorithm and also with many other sources which are directly or indirectly connected with

other information sources. We have to update the links and information time to time.

**Priority**  - 5

**Risk** - 2

**Request Id** - Genome/Rid/6

**Title**  - Maintenance Functionality/Site maintenance

**Description** - Site maintenance is very important for this purpose and we have to perform a great deal of work for this. Time to time we have to check our site information. As our application is an educational application, so it’s very important to keep the informations up to date.

**Priority** - 2

**Risk**  - 1

**Request Id** - Genome/Rid/7

**Title**  - Maintenance Functionality/Upgrade

**Description** - We will upgrade our application time to time, with necessary informations, necessary apps and new options and modifications along with new and newer technologies including development of graphical interface of our app. So It’s one of the most important priority of us.

**Priority** - 1

**Risk**  - 2

**3.1.3 Graphical User Interface**

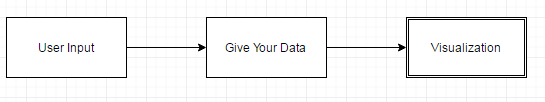
**Request Id** - Genome/Rid/8

**Title**  - Graphical User Interface/Design Architecture

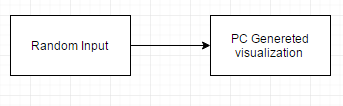
**Description** - The most important feature of our application is attractive graphical level interface through which people communicate with the device. We used java applet and awt packages for swiftness, including java swing, javafx and other external graphics. We have a very bold and interactive interface to communicate. Besides that we are keeping multiple operation to change the working environment and flavour through custom settings according to the choice of user. And the interface is going to upgraded time to time from our server to keep a balance with the user’s choice. There will be option which will define the referenceable choice and how the choice is selected on basis of MAX ONE problem.

**Image 4 - represents the first Interactive displays**

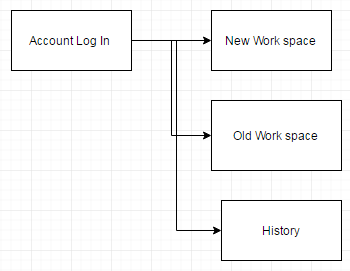
**Image 1,2,3 - represents different level of Interaction**

****

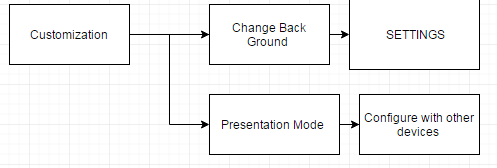
**Image 6 - User Level Interaction (User Choice)**

****

**Image 7 - User Level Interaction (Random Choice)**

****

**Image 8 - Account Information**

****

**Image 9 - Customization**

**These are basically the basic level work depth level. There will be multiple level work and process level in the flow.**

**Priority** - 1

**Risk**  - 1

**3.1.4 Interfaces to External Systems**

**Request Id** - Genome/Rid/9

**Title**  - External Systems/Others

**Description** - Genome is suitable for attaching with multiple devices. Specially for presentation or wide screen view purpose. There will be two ways. First way is to connect it directly with the device and smartphone and operate from other devices or doing some operations in the smart phone and then making downloadable files and using them as presentation. So for the first option genome will have a interface to communicate with other external devices with smart phone.

**Image 5 - gives a demonstration of this feature.**

**Priority** - 2

**Risk** - 2

**3.2 Usability**

**3.2.1 Language**

**Request Id** - Genome/Rid/10

**Title** - Usability/Language

**Description** - Genome supports two languages now. English and Bangla. As English is one of the most usable communication language now a days. Genome full functional language will be English. Genome will also have Bangla as it’s second language. It will also be fully functional.

In future team will work for adding more languages to it.

**Priority**  - 4

**Risk**  - 3

**3.2.2 History**

**Request Id** - Genome/Rid/11

**Title** - Usability/History

**Description** - Genome will save the history of the user. It will be a custom operation and viewers will select how many last use history will be saved and how they will be accessed.Generally last 5 times use will be saved with secure authorization. And to have more from the past history additional memory space must be permitted by the user and from that they can backtrack the history and get the outcome.

**Priority** - 2

**Risk** - 1

**3.3 Reliability**

**3.3.1 Availability**

**Rid -** Genome/Rid/12

**Title** - Reliability/Availability

**Description** - Our device will be available on google play store. So it will be available to all the people. Except that we will offer our product to the educational institutions and other research groups who need them. By that we can make our product available to all other people.

**Priority**  - 1

**Risk**  - 4

**3.3.2 Error**

**Request Id -** Genome/Rid/13

**Title**  - Reliability/Error

**Description** - Error can be one of the most important factor while launching the product. So, from that perspective we are following much and much cautiousness regarding this. So before launching our product we will bug test of the software using **Lean Testing, Jira, Mantis, Trac, Redmine, ALM** and **FogBugz .** After and only after getting satisfactory result from these softwares we will launch our product. Till now we have got a good result.

**Priority** - 1

**Risk**  - 4

**3.3.3 Error Handling**

**Request Id -** Genome/Rid/20

**Title** - Reliability/Error Handling

**Description**  - If there any type of error occur while running the software , through the agreement with the user and account with the server, users can make know us about their problem and under 24 hours we will solve the problem and again send request to the user through mail and necessary formation. There we will send a link in the mail and by clicking the mail, the user will synchronize with the mail and new software will be automatically updated with the errotic ones.

**Priority - 1**

**Risk - 1**

**3.4 Security**

**Request Id** - Genome/Rid/14

**Title** - Security/Authorization

**Description** - Our users can have personal account on our software and through them they can keep their using data safe. And each account can have encapsulation and owner authorization process. They can give others authorization to the program.

**Priority**  - 4

**Risk**  - 4

**3.5 Hardware**

**Request Id**  - Genome/Rid/15

**Title -**  Hardware

**Description -** Our program do not need any extra hardware dependency. The platform where the software will run (EX - smart devices supporting android operating system is enough to maintain the program data flow. There is not needed any extra hardware dependency. The existing device’s usage hardware mechanism would be enough for that purpose.

**Priority**  - 1

**Risk** - 4

**3.6 Performance**

**Request Id** - Genome/Rid/16

**Title**  - Performance

**Description**  - Performance factor is one of the most vital thing that plays a role while launching the product. So before launching our product we will check our performance rate from both beta version and also by special performance measuring softwares. For example , by using beta version of our software users can give their valuable comments which will play a role while determining the outcome, and also by special softwares like **webload, loadUI NG Pro, Apica** and **LoadView.** These will determine the performance rate of our softwares.

Besides - Following factors will be handled cautiously while determining the performance factors.They are - **Load Testing, Stress Testing, Soak testing ,Spike testing Configuration testing.** Performance rate is now 3.6 out of 5.0 and the process is being performed to make it better.

**Priority**  - 1

**Risk**  - 1

**3.7 Operating System**

**Request Id** - Genome/Rid/17

**Title** - Operating System/Android

**Description**  - Our totally application is based on android operating system.

**Android** (stylized as **android**) is a [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) developed by [Google](https://en.wikipedia.org/wiki/Google), based on the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and designed primarily for [touchscreen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer). Android's [user interface](https://en.wikipedia.org/wiki/User_interface) is mainly based on [direct manipulation](https://en.wikipedia.org/wiki/Direct_manipulation_interface), using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a [virtual keyboard](https://en.wikipedia.org/wiki/Virtual_keyboard) for text input.

Our total software is based on these intervallistic methods of android and applicable to it.

The core of this software will be based on android. So the devices which will support it will be applicable to use our software.

We choose android as our best option, because android is one of the most easiest, famous and widely used OS now, By building something on the basis of android we can actually reach more and more people than using any other OS. And statistically it is found that the use of android will not lessen rather it will increase in future times. Basically our program will run on android operating system with minimum requirement of Ice Cream Sandwich 4.0 with API level of 25.

**Priority - 1**

**Risk - 1**

**3.8 Deployment**

**Request Id** - Genome/Rid/18

**Title**  - Deployment/Upgrade

**Description** - Genome team will upgrade the software each year. After each year we will release a better version and ask our users to upgrade them. From which they will synchronize with their compatible version with the newer version.

**Priority** - 2

**Risk**  - 2

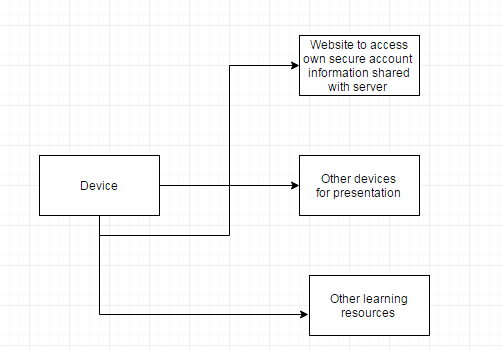
**3.9 Legal Copyrights, licensing requirements, online user documents**

**Request Id** - Genome/Rid/19

**Title**  - Core Documents/storage

**Description** - Our software is totally for educational purpose. But sharing the software will be un approved. And for this reason while downloading as there will be little bit of cost issue there will be an agreement. Besides, another issue which is important they can not share this without authority permission. So, after first login there will be issued an agreement on which the user can perform. We are going to keep these issues in mind.

**3.10 Connection among the parts**



**Image 10 - Various Device/Medium Interconnections**

**4 Change Management Process**

Identify the change management process to be used to identify, log, evaluate, and update

the SRS to reflect changes in project scope and requirements. How are you going to

control changes to the requirements. Can the customer just call up and ask for something

new? Does your team have to reach consensus? How do changes to requirements get

submitted to the team? Formally in writing, email or phone call? How to keep track the information which are daily updated? How to track the sources of new information. How to upgrade the software and information according to the need of users.

**5 Document Approvals**

Identify the approvers of the SRS document. Approver name, signature, and date should be used.

**6 Supporting Informations**

The supporting informations make SRS easier to use. They are -

1. Table of Contents
2. Index

**7 References**

1. Wikipedia
2. <http://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
3. Android Operating System Brief Introduction
4. Tutorials Point

**8 Tools**

1. Microsoft Word
2. Microsoft Paint
3. [www.draw.io.com](http://www.draw.io.com)
4. [www.Google.com/document](http://www.google.com/document)

Thank You.