BrightBorn

Software Development Plan

Zeynep Ovgu YAYCI & Ugur DURA & Mustafa AYKAC & Cem OZCAN & Burak ERDOGAN

12/05/2020

Contents

1. Overview
Problem Definition

We are living in a globalized world, and as a result of this, there is an increase in the number of inter-race couples. Thus, the gene combinations that cause genetical diseases also increased. Since it is a long process to find treatment for genetic diseases, we are aiming to prevent them from beginning. In order to do that we combined software technology and genetical knowledge together to create an algorithm to find the possibility of these genetical diseases.

Background Information

After the discovery of independent assortments and segregations of genes by Gregor Mendel, modern genetic have gained momentum since the 19th century. Nowadays scientists developed this knowledge into advance. At present, while a couple is planning to marry, they check their blood types to prevent later complications such as RH disease. We are planning to bring forward this process and estimate the couple's possible offspring by combining the features of software computing and genetical knowledge.

Objectives

- Generating synthetic genomes
- Taking genetic information from partners
- Checking the similarities between participants genome and ex clinical results
- Determining mutations in participants genome
- Simulating crossing over
- Creating possible offspring of both participants
- Giving the possibility of participants' children having a genetical disease

Scope

During our design, as a team, we decided to scale down the size of the human genome. The reason being is the human genome contains 4.5 billion base pairs and around 20,449 different genes (Ensemble genome browser release). It may create some implantation difficulties. Instead of checking each one of them, we just focused on the common genes which can generate major problems such as hemoglobin beta gene (HBB) or cellular tumor antigen p53. We decided to generate a synthetic genome which we defined the genes inside already. We choose GRCh38 reference genome as our template and we add inside of this template Cystic fibrosis transmembrane conductance regulator gene (CFTR-In the case of mutation of this gene causes

Cystic Fibrosis), Hemoglobin beta gene (HBB-In the case of mutation of this gene causes Sickle cell anemia), Oculocutaneous albinism II gene (OCA2-In the case of mutation of this gene causes Albinism), Phenylalanine hydroxylase gene (PAH-In the case of mutation of this gene causes Phenylketonuria), Huntington gene (HTT-In the case of mutation of this gene causes Huntington disease) and Tumor protein p53(TP53- In most cancer cells, this gene found as damaged). We have ongoing discussions about adding one more chromosome which also includes some basic genes which define eye or hair color or stature near this synthetic chromosome which contains potential genetical disease genes. While we were working on the synthetic genome, we added telomer sides to the both ending part, some noncoding parts and also crossing over sites into the genome to increase accuracy. Besides, instead of using basic algorithms which check only the difference with the reference genome, we decided to combine clinical results to increase precision. The reason for this is examining ex clinical results with software computation is easier and advisable when we compare to understanding whole signaling pathways, genetic correlation and originate connection between them.

The program starts the process by taking genetic information from partners. Then, the program checks against differences between reference genes. If the result is unhealthy, the program starts to check against ex clinical results to find a similarity between participants genome and ex clinical results. After the determination of the mutations of partners, programs start to simulate the crossing over to create possible offspring of both participants. In the final stage, the program combines these offspring and examines the new possible genomes with the reference genome to give the result of their child according to the possibility of getting a genetic disease.

This Project may lead the way for designer baby tools. In case of participants want to have a baby, they will have an opportunity to change damaged parts of their offspring to prevent later complications.

2. HIGH-LEVEL FUNCTIONALITY

2.1) Functional Requirements

- Users shall be able to register and login to the system with their ID and password via a log in page
- There should be a 'Forgot my password' option
- User shall be able to upload one text file for each parent which includes their DNA code in 'fatsa' format
- The staff shall have their ID and password
- The program shall be able to implement the crossing-over process
- The output should be given as a report including possibilities of desired diseases
- In result report the mutant parts of the gene should be shown to the user with red color
- During maintenance, the system should warn the users and should be closed for them
- The staff shall be able to access to the database and may add new clinical data

2.2) Nonfunctional Requirements

- Run time should not exceed 2-3 min
- The program shall be based on local storage and database design should allow the changes
- A user's data should only be accessed by the same user's ID and password
- The program should be runnable by 500-2250 processors
- The system shall be accessible for multiple users at the same time
- The design and report should be easily understandable with general knowledge and user-friendly

3. STAKEHOLDERS

We have six stakeholders in our project:

a) **Patients:** They will check the genetical disease possibility and make decisions on having children

b) **Doctors:** They will have easier access to their patients, and they won't have to find results themselves

c) **Researchers:** They can examine inter race marriage effects on human breeding easier

d) **QIAGEN:** Owner of our reference genetical disease database

e) **Hospital:** They can manage the genomic disease department easier

f) **Staff:** Staff needs to make maintenance for upcoming updates and fixes

4. PROJECT STAFFING

There will be five types of project staff: a software project manager, requirements engineer, a lead designer, a tester and a project developer.

Software Project Manager: Zeynep is responsible for planning, scheduling, and reporting the project; managing the team members and organizing the tool selection etc.

Requirements Engineer: Ugur is responsible for managing the stakeholders, analyzing, and documenting the software requirements.

Lead Designer: Burak is responsible for designing the user interface, GUI and connection between SQL and the program.

Tester: Mustafa is responsible for testing all the positive and negative scenarios.

Project Developer: Cem is responsible for coding according to requirements engineer's and lead designer's wants. Works with manager in developing the project plan and schedule

5. SOFTWARE PROCESS MODEL

5.1) Necessary Needs from The Organizational Process

- All the members should take Python course in their related area.

- Every end of the week there must be a meeting to discuss to what everyone has done.

- The genetic information that is going to be used for comparing other gens must be from a real person.

- Testing must be done after every implementation.

- Everyone must update their implementations on GitHub.

- Due to our project complexity, the estimated time for project is approximately between 3 to 4 months

- Risks are too much to make any wrong calculations in the probability of sickness.

5.2) Unnecessary Needs from The Organizational Process

- We do not need any devices to embezzle our project members.

- Pair programming will not be needed.

- Frequently contacting with the customers will not be needed due to pre-determined requirements.

- Due to simple design, there will not be much time given for designing processes.


5.3) Software Process That We Use: V Model

In V model the process executes in a sequential manner. For each development activity there is a corresponding testing activity. Before starting the next stage, the previous stage needs to be completed.

In this V shape, one side contains the design phase and other side contains the corresponding testing phase.

It could be used in small and medium sized projects.

The requirements should be well-defined before starting the software process.
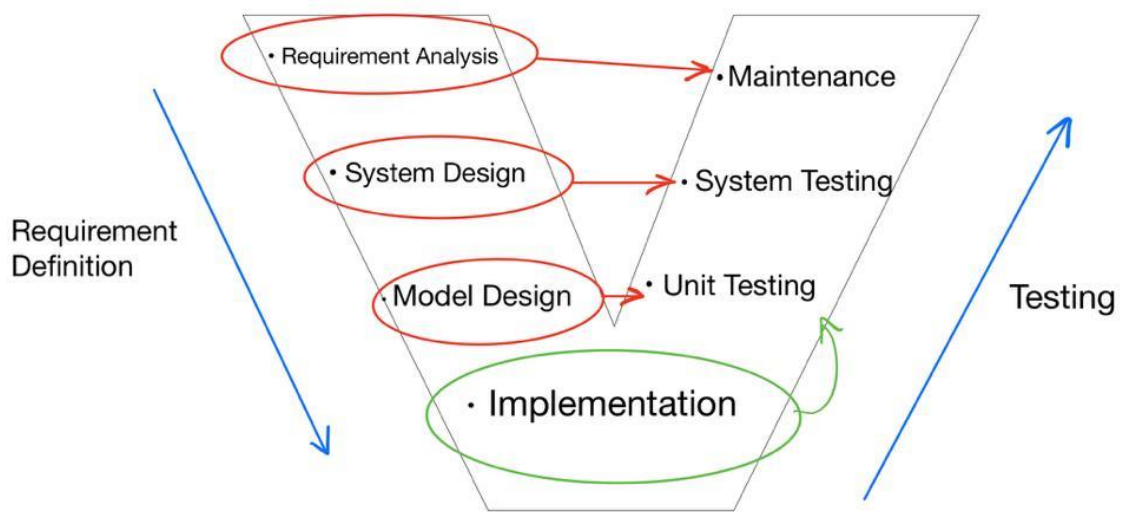
Figure 1: The V Model

**5.4) Reasons to Choose This Model:**

- The project requirements are clear
- Since the program is health related, each stage needs to be carefully designed and tested
- Our project is a medium sized project, so the V model is appropriate
- Since our projects has a dynamic database, after all the stages are finished; maintenance needs to be continuous
- Since our project has a high-risk potential; this model will work better than some other models
- In our project technical resources that we need are already available

6. SCHEDULE AND EFFORT

		Starting Time	Finishing time	Duration
1	Requirement Gathering	Starting Time	Finishing time	Duration
2	Background Reading	10.02.2020	13.02.2020	3
3	Defining Questions and Problems	13.02.2020	14.02.2020	1
4	Literature and review research	14.02.2020	15.02.2020	1
5	Arrange Meetings	16.02.2020	17.02.2020	1
6	Conduct Meetings	17.02.2020	18.02.2020	1
7	Collecting outputs from reviews	18.02.2020	19.02.2020	1
8	Prepare Reports	19.02.2020	20.02.2020	1
9	Analyze and Define Requirements	20.02.2020	25.02.2020	5
10	Acceptance Testing	25.02.2020	27.02.2020	2
11	Design			
12	UML Design Creation	4.03.2020	7.03.2020	3
13	Architectural Design	7.03.2020	14.03.2020	7
14	Graphical Design	14.03.2020	28.03.2020	14
15	Animation Design	14.03.2020	28.03.2020	14
16	Database Design	14.03.2020	21.03.2020	7
17	Interface Design	14.03.2020	28.03.2020	14
18	Create Design Specifications	28.03.2020	1.04.2020	4
19	Design Complete	1.04.2020	8.04.2020	7
20	Design Testing	8.04.2020	10.04.2020	2
21	Coding			
22	Development of System Modules	10.04.2020	8.05.2020	28
23	Unit Testing	8.05.2020	10.05.2020	2
24	Database Development			
25	SQL	8.05.2020	22.05.2020	14
26	GUI and Interface Development			
27	Maya	8.05.2020	22.05.2020	14
28	Qtdesigner	8.05.2020	22.05.2020	14
29	Integrate System Modules	22.05.2020	1.06.2020	10
30	Perform Initial Testing	1.06.2020	2.06.2020	1
31	Development Complete	2.06.2020	3.06.2020	1
32	GUI Testing	3.06.2020	4.06.2020	1
33	Overall System Testing	4.06.2020	7.06.2020	3

Figure 2: Schedule of the project
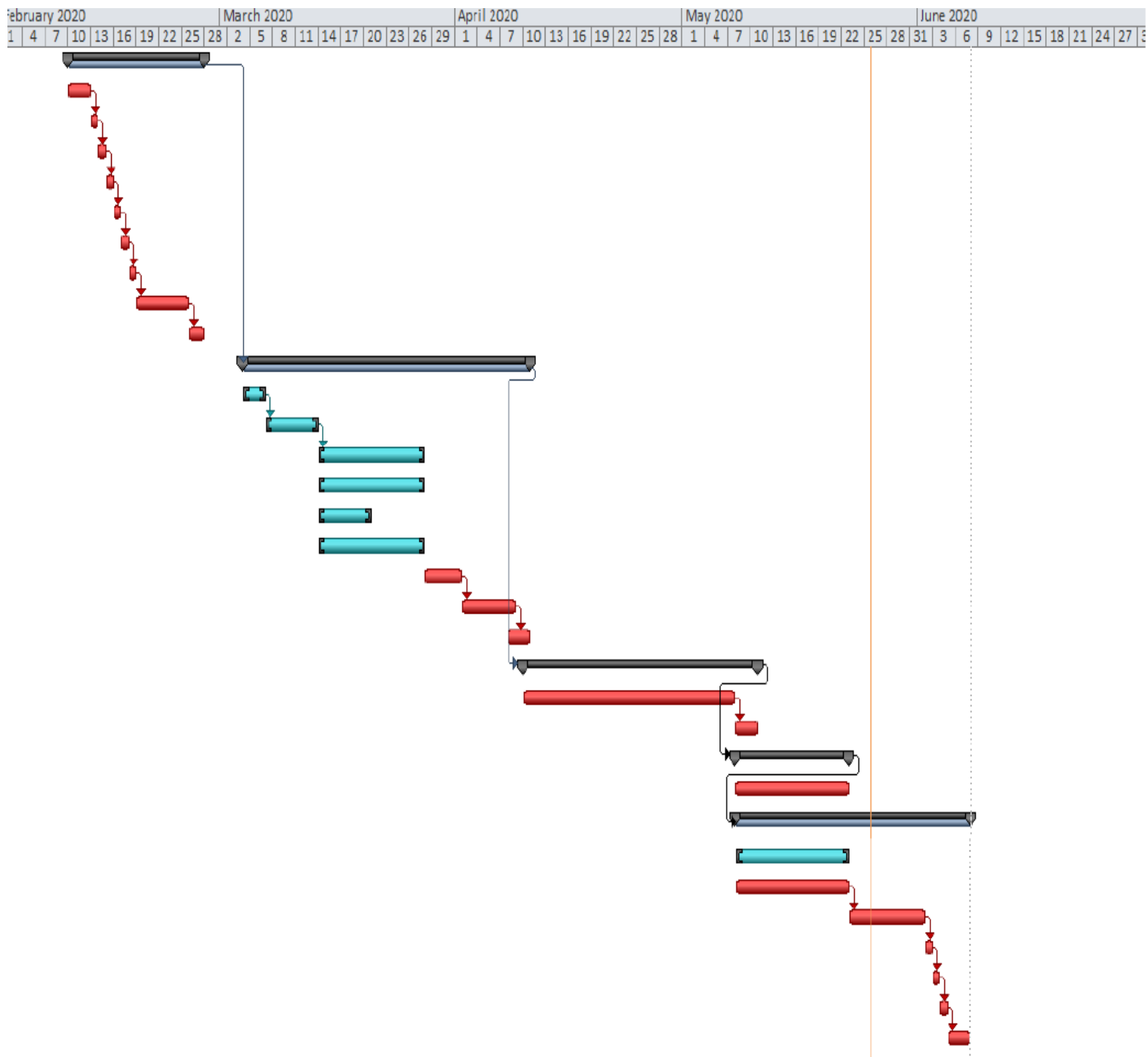
Figure 3: Gantt Chart

7. MEASUREMENTS

In this project we need to use six different types of measurements:
- **Dataset Storage:** There will be an initial limit for our dataset storage thus each week Cem will check it and if it is close to the limit, he will increase the limit.
- **Data Collection:** The collection of the genomic data will be check by Ugur continuously by searching on literature.
- **Performance and Evaluation:** Accuracy will be based on possibility calculations by Zeynep and she will calculate the percentage error after every iteration.
- **Response Time:** The execution time will be checked at the end of every execution by Mustafa.
- **Code Size Measurement:** The lines of code will be checked by Burak after every maintenance.
- **Effort Distribution:** Ugur will check the research time after every week in hours and report it.

8. PROJECT RISKS

8.1) Likelihood Risk List:

LIKELIHOOD RANK	RISK DESCRIPTION
1	**Data Accessibility** – If the companies holding the data required for comparison with other genes restrict access to this data, we may not be able to obtain sufficient data thats why we need to search for alternatives.
2	**Hardware Limitation** – During the computations for marking mutant genes and possible outcome, virtual space might not be enough for certain computations. It should be tested and optimized on every step.
3	**Complex Calculations** – According to accuracy of reports; complex possibility calculations lead software to generate weak spots thats why it error should be checked on every iteration.
4	**Tools** – The team needs to learn new tools to support the project otherwise the required sofware tools can be formed according to the team members skills.
5	**Debugging** – Due to the complex calculations required for the project. Errors can lead to hard debug section for team members thats why we can search for new algorithms that can make our calculations less complex.
6	**Training** – We have limited time to finish this project; so the team members should train in a short period of time.
7	**UI Complexity** – User interface should not create any dilemma thus it should be clear to understand by users.

8.2) Impact Risk List:

IMPACT RANK	RISK DESCRIPTION
1	**Complex Calculations** – According to accuracy of reports; complex possibility calculations lead software to generate weak spots thats why it error should be checked on every iteration.
2	**Hardware Limitation** – During the computations for marking mutant genes and possible outcome, virtual space might not be enough for certain computations. It should be tested and optimized on every step.
3	**Training** – We have limited time to finish this project; so the team members should train in a short period of time.
4	**UI Complexity** – User interface should not create any dilemma thus it should be clear to understand by users.
5	**Tools** – The team needs to learn new tools to support the project otherwise the required sofware tools can be formed according to the team members skills.
6	**Debugging** – Due to the complex calculations required for the project, any of the errors can make debugging hard for the team.
7	**Data Accessibility** – If the companies holding the data required for comparison with other genes restrict access to this data, we may not be able to obtain sufficient data thats why we need to search for alternatives.

8.3) Combined Risk List:

LIKELIHOOD RANK	IMPACT RANK	COMBINED RANK	RISK DESCRIPTION
3	1	4	**Complex Calculations** – according to accuracy of reports; complex possibility calculations lead software to generate weak spots
2	2	4	**Hardware Limitation** – During the computations for marking mutant genes and possible outcome, virtual space might not be enough for certain computations. It should be tested and optimized on every step.
1	7	8	**Data Accessibility** – If the companies holding the data required for comparison with other genes restrict access to this data, we may not be able to obtain sufficient data thats why we need to search for alternatives.
4	5	9	**Tools** – The team needs to learn new tools to support the project otherwise the required sofware tools can be formed according to the team members skills.
6	3	9	**Training** – We have limited time to finish this project; so the team members should train in a short period of time.
7	4	11	**UI Complexity** – User interface should not create any dilemma thus it should be clear to understand by users.
5	6	11	**Debugging** – Due to the complex calculations required 11ort he project. Any of the errors can make debugging hard for the team.

9.SOFTWARE TOOLS

Three project tasks which require software tool support: Database Management, 2D-3D Animations and User Interface.

9.1) Software Tools for Database Management:

9.1.1) Tool Cost / Training / Functionality Data:

Tool	IBM DB2	Oracle RDBMS	Microsoft SQL Server	Adabas
Cost	$1850	$5800	$3717	$10000
Training Days	6	4	2	7
Functionality	64	50	85	24

9.1.2) Normalized Tool Cost / Training / Functionality Data:

Tool	IBM DB2	Oracle RDBMS	Microsoft SQL Server	Adabas
Cost	18.5	58	37.17	100
Training Days	85.71	57.14	28.57	100
Functionality	75.29	58.82	100	28.24

9.1.3) Normalized Tool Graph:
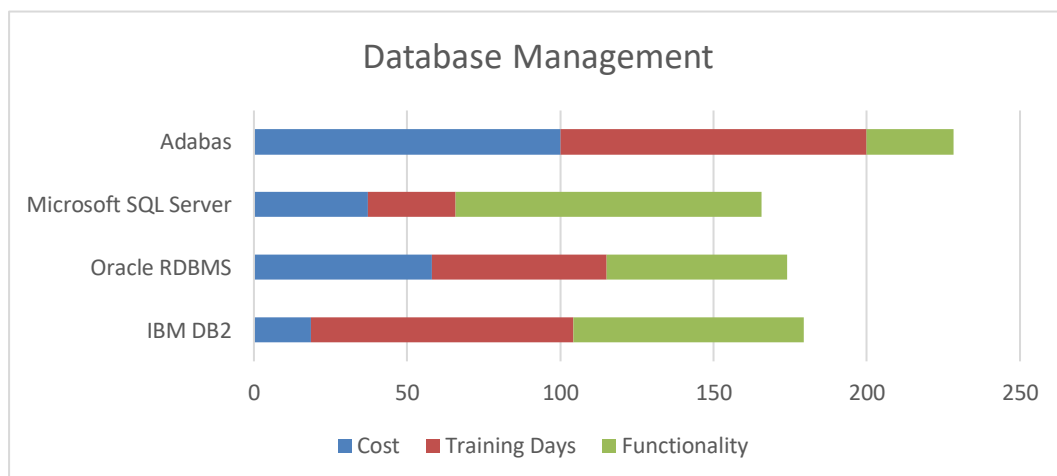
9.1.4) Which tool has been selected? Why?

We decided to go on with Microsoft SQL server. The reason for this is;

- It provides high functionality when we compare it with other options.
- Low training days is crucial for our Project.

12

9.2) Software Tools for 2D-3D Animations:

9.2.1) Tool Cost / Training / Functionality Data:

Tool	Maya	Blender	3ds Max	Modo
Cost	$1500	$0	$1545	$1799
Training Days	2	3	5	1
Functionality	25	32	50	45

9.2.2) Normalized Tool Cost / Training / Functionality Data:

Tool	Maya	Blender	3ds Max	Modo
Cost	83.37	0	85.88	100
Training Days	40	60	100	20
Functionality	50	64	100	90

9.2.3) Normalized Tool Graph:
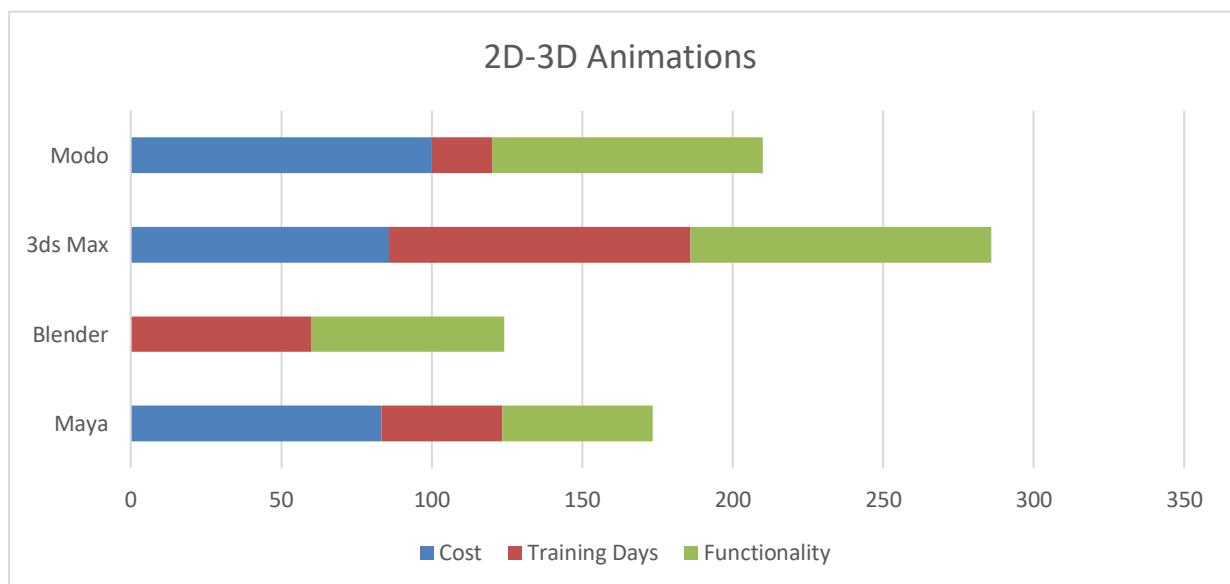
9.2.4) Which tool has been selected? Why?

We decided to go on with Modo. The reason for this is ;

-When we compare Modo with other options , first priority for our Project is time.

-Modo gives us high functionality with low training days.

9.3) Software Tools for User Interface:

9.3.1) Tool Cost / Training / Functionality Data:

Tool	Django	QtDesigner	Java Swing	Kivy
Cost	$150	$0	$0	$0
Training Days	4	1	5	2
Functionality	45	50	20	45

9.3.2) Normalized Tool Cost / Training / Functionality Data:

Tool	Django	QtDesigner	Java Swing	Kivy
Cost	100	0	0	0
Training Days	80	20	100	40
Functionality	90	100	40	90

9.3.3) Normalized Tool Graph:
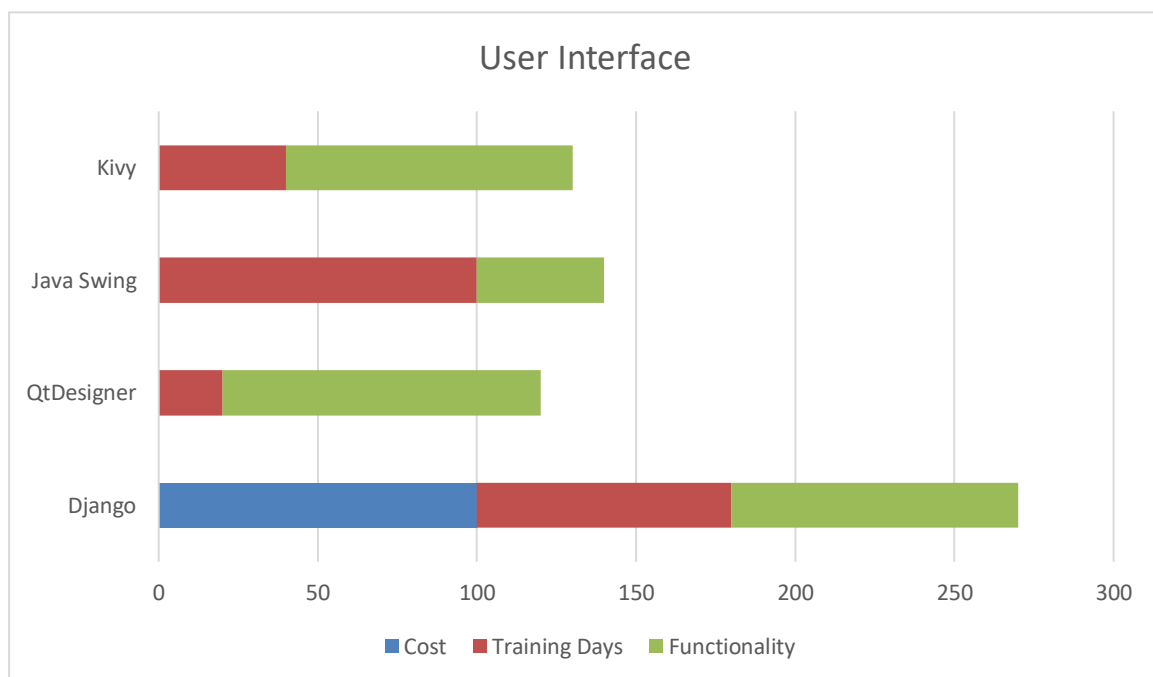
9.3.4) Which tool has been selected? Why?

We decided to go on with QtDesigner. The reason for this is;

- QtDesigner provides us available documentations in wide range and it has low cost and low training days.
- It fits for costumers' requirements about project.

10. PROJECT NEEDS

10.1) Software Needs:

#	SOFTWARE NEEDS	DESCRIPTION
1	Microsoft SQL Server	SQL lates version (version is 15.0. 2000.5) – Cem ÖZCAN and Uğur DURA are responsible for installing and upgrading
2	Operating System	Windows 10 – Mustafa AYKAÇ is responsible for installing and upgrading
3	Modo	13.0v1 – Burak ERDOĞAN is responsible for installing and upgrading
4	QtDesigner	QtDesigner 5.6.0 – Burak ERDOĞAN and Mustafa Aykaç are responsible for installing and upgrading
5	Office 365	2019 – Zeynep Övgü YAYCI is responsible for installing and upgrading

10.2) Hardware Needs:

#	HARDWARE NEEDS	DESCRIPTION
1	Display Screen	AOC 27" C27G1 144 Hz 1ms HDMI DP FreeSync / G-Sync Full HD - Used to see the results
2	Disk Space	SanDisk Ultra 3D 2TB 560MB-530MB/s Sata 3 2.5" SSD (SDSSDH3-2T00-G25) - To store our genetical information
3	Display Card	MSI GeForce RTX 2080 Ti GAMING X TRIO 11GB GDDR6 352 Bit Ekran Kartı -To take inputs and process 3D data efficiently
4	Processor	INTEL COFFEELAKE CORE I9 9900KF 3.6GHz 1151P 16MB – To compute mathematical calculations

10.3) Support Needs:

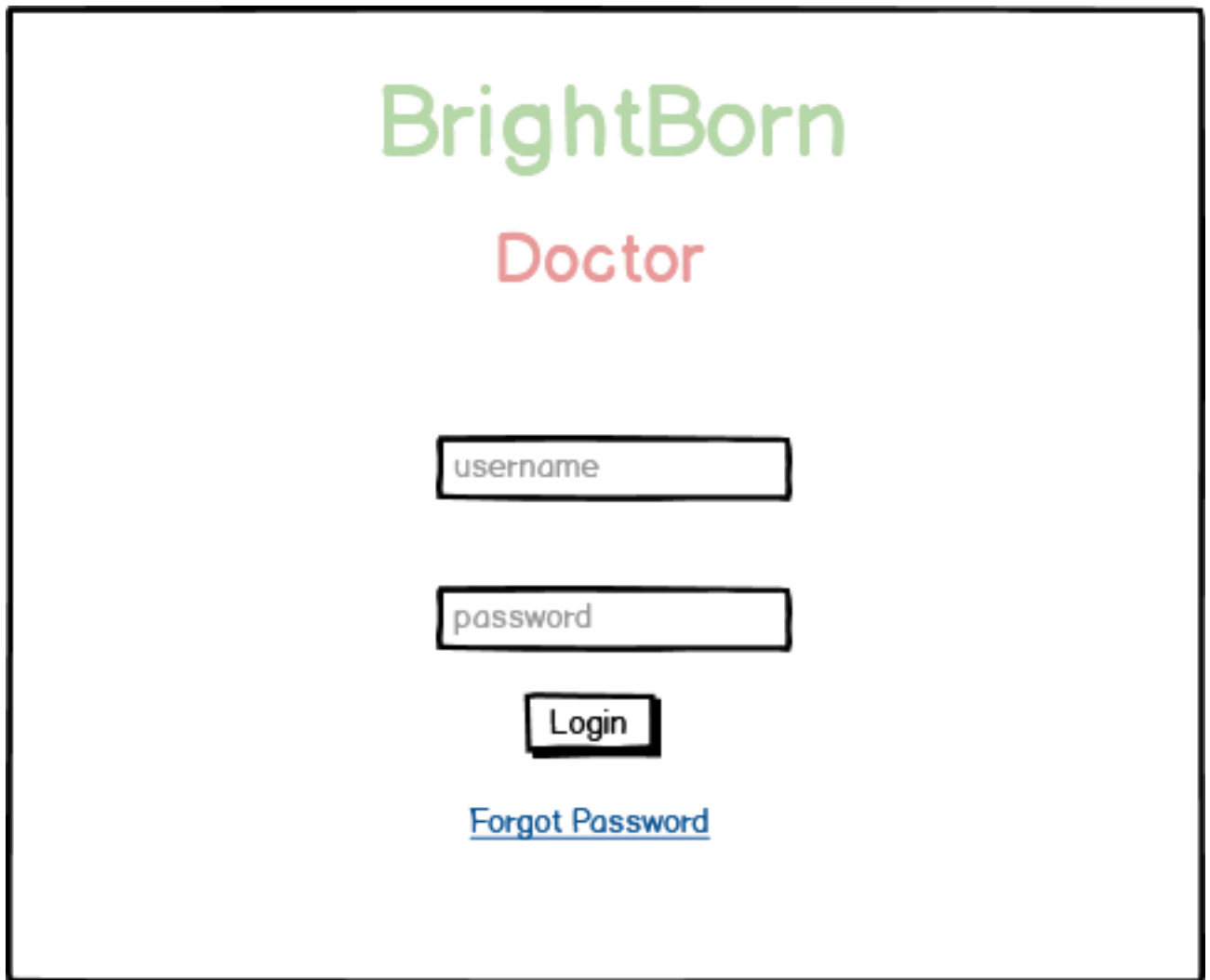| #	SUPPORT NEEDS	DESCRIPTION
1	Genetical Information	The data we need to get from other databases
2	Genetical Knowledge	While we are designing algorithms for possible breadings we need to construct according to this knowledge
3	Mathematical Knowledge	We need to implement possibility calculations in order to reach a result.
4	Clinical Results	We can get these data with using ex-clinical research reviews to increase accuracy

11. GRAPHICAL USER INTERFACES
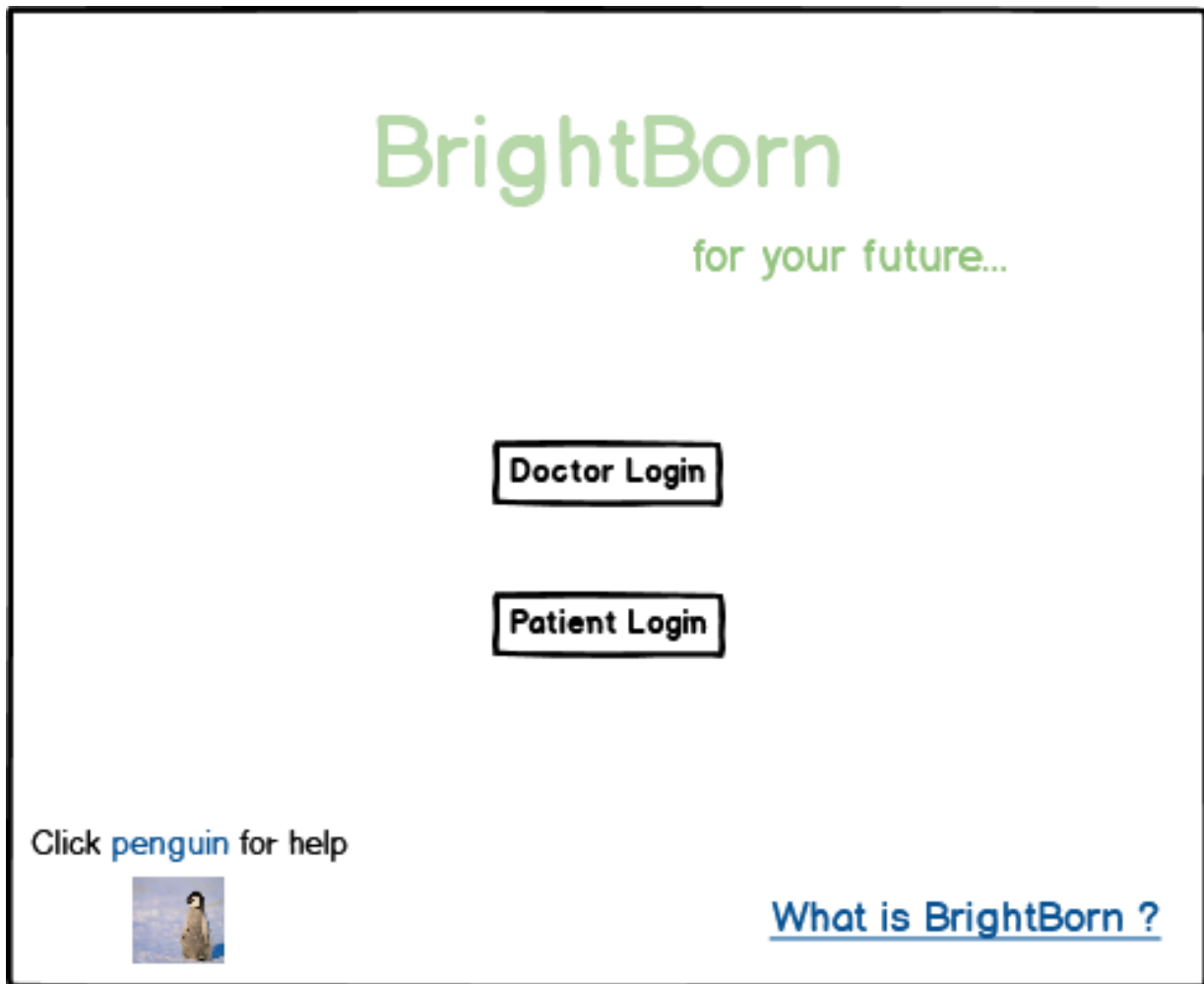
Figure 1: Main Page

In the main page:

- The Doctor Login option will open the Doctor Login page (Figure 2)

- The Patient Login option will open the Patient Login page (Figure 4)

- Penguin image will open the Help page (Figure 7)

- 'What is BrightBorn?' option will lead to What is BrightBorn page (Figure 8)

Figure 2: Doctor Log-in Page

In the Doctor Log-in page:

- After entering username and password log-in button will lead to Doctor page (Figure3)

- Forgot Password option will send to the Help page (Figure 7)

Figure 3: Doctor Page

In Doctor page:

- Check Report option will send the user to Report Page (Figure 5)

Figure 4: Patient Log-in Page

In Patient Log-in Page:

- After entering a valid Report ID; Check Report button will send the user to Report
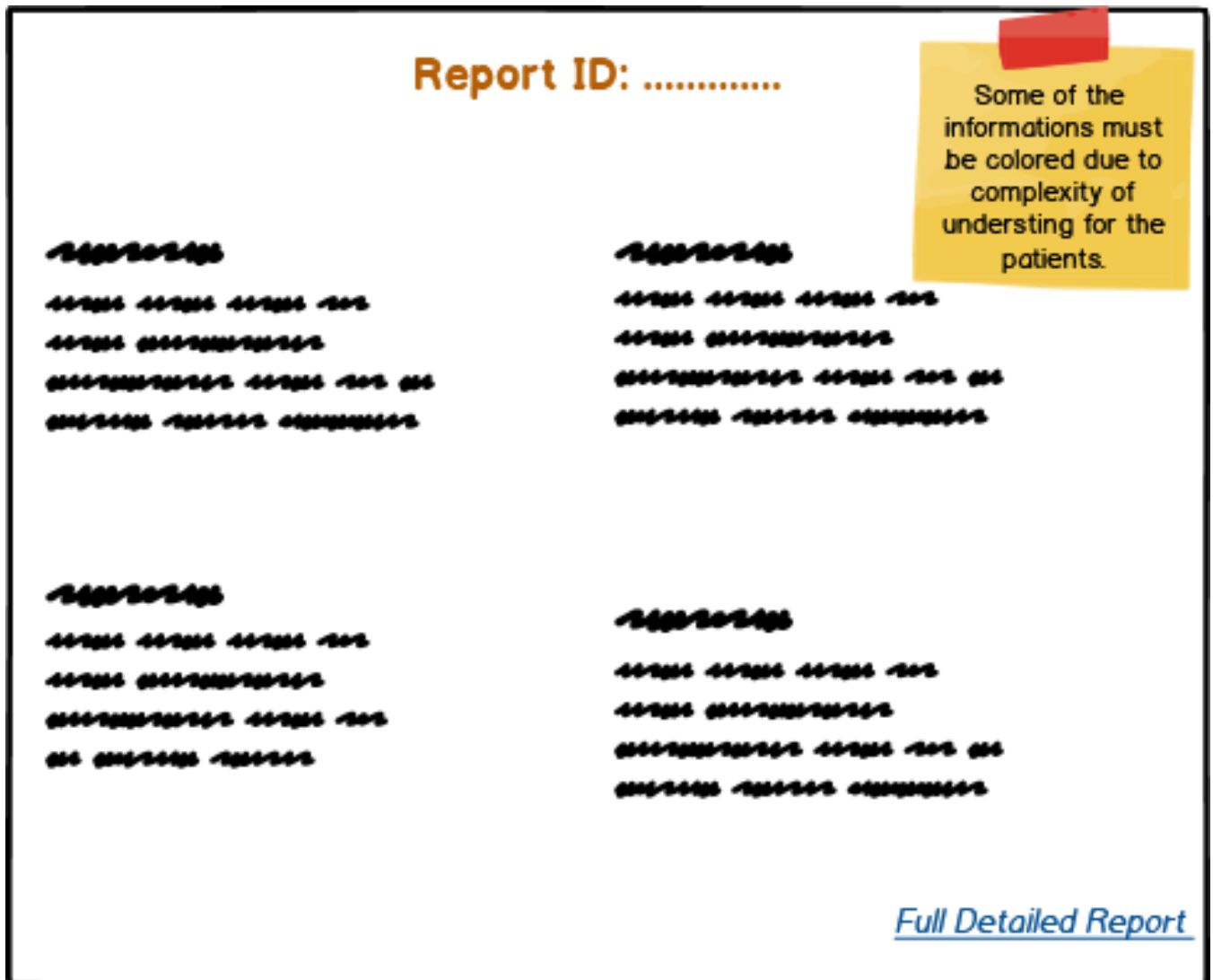 Page (Figure 5)

Figure 5: Report Page

In the Report Page:

- Full Detailed Report Option will send the user to Detailed Report Page (Figure 6)
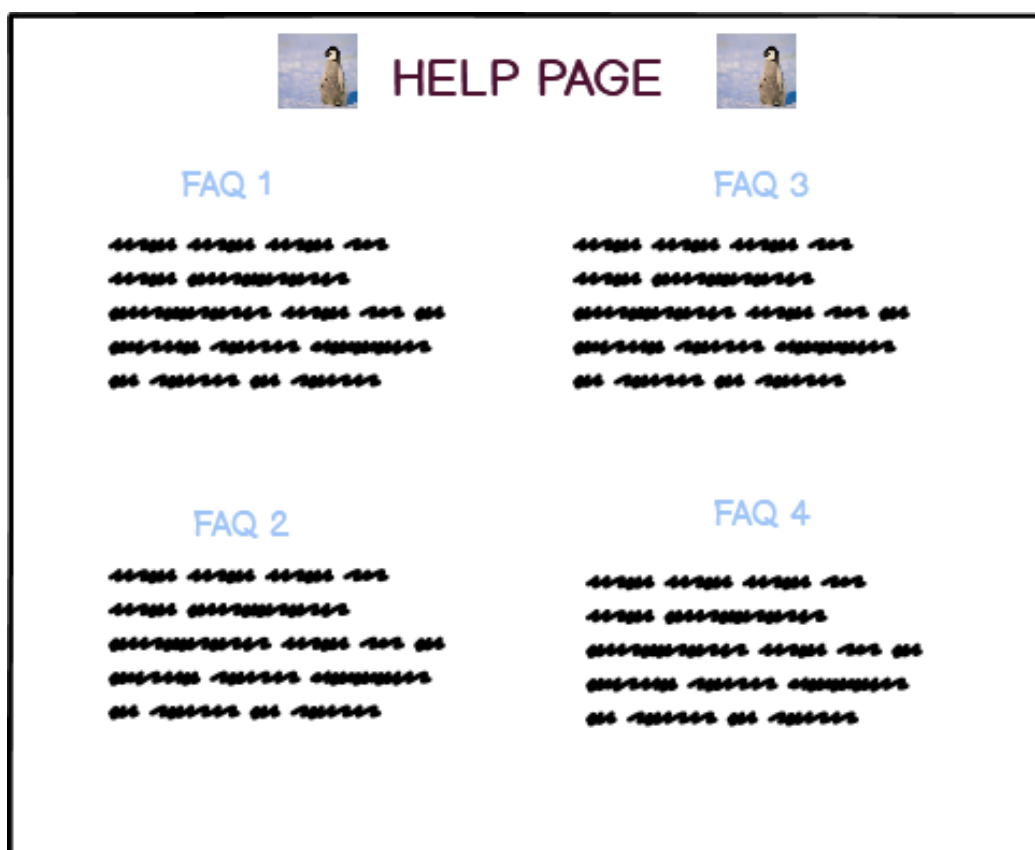
Figure 6: **Detailed Report Page**

Figure 7: Help Page

Figure 8: What is BrighBorn?

12. CONCLUSION

This document gives a clear idea about the project BrightBorn. It explains the overview which includes the problem definition, background information, objectives and the scope of the project, high-level functionality which includes functional and nonfunctional requirements, stakeholders, project staffing, software process model that the project uses and the reasons to chose it, schedule and effort and the Gantt chart, measurements, project risks, software tools, project needs and GUI. The software manager who is planning to follow this SDP need to be aware of possible risk that mentioned before. According to possible risks, finding and accessing on ex clinical results has a key role over the course of development and advancement of BrightBorn. If everything goes according to this plan, it will be a great project!