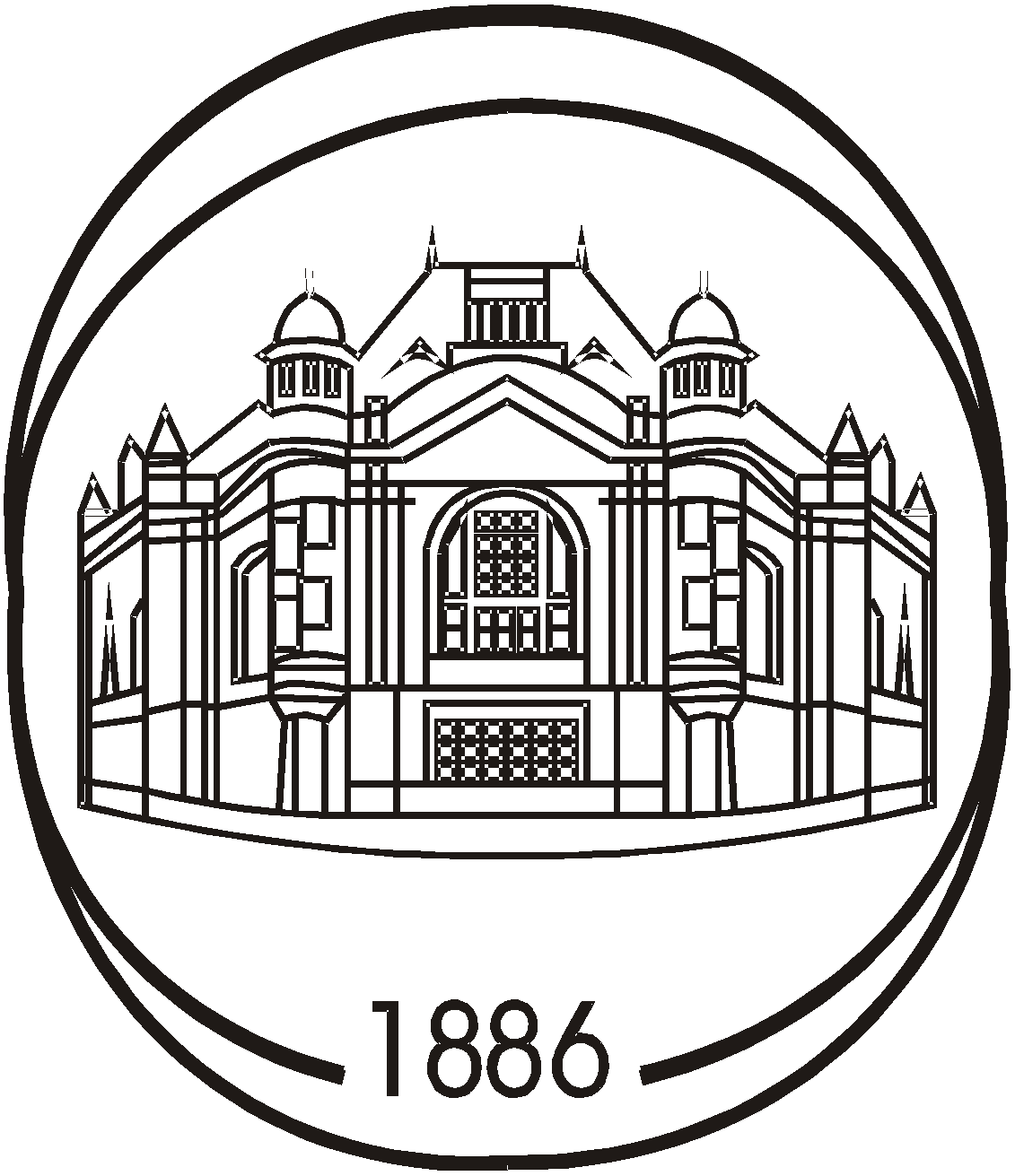
MINISTRY FOR EDUCATION AND SCIENCE OF RUSSIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SAINT PETERSBURG ELECTROTECHNICAL UNIVERSITY

«LETI»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**APPROVED**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date, signature

Personal Information Encryption

Project title

PROJECT MANAGEMENT PLAN

PMP\_PIE\_00.30

        Electronic & Paper

13 pages

St. Petersburg / Санкт-Петербург 2019

**HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version # №** | **Author** | **Data** | **Short description / Причина** |
| 00.10 | *Salwan* | *18.03.19* | *The first version* |
| 00.20 | *Salwan* | *25.03.19* | *Detail requirements, gant diagram* |
| 00.30 | *Salwan* | *01.04.19* | *Architecture, prepared for review* |

**Project team**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Name** | **Group** | **Project role** | **Contact Information** |
| 1 | Salwan  Chasib | 4300 | PM, test lead, dev, CM | toniandthemachine@gmail.com  mob.+7 9872340196 |

**Technical expert – coacher**

**\_\_\_\_Salwan Luay Chasib\_\_\_ \_\_\_\_toniandthemachine@gmail.com\_\_\_**

**Full name email**

**Teacher**

**Filippov Evgeny Vasilyevich** [**e.philippov.leti@mail.ru**](mailto:e.philippov.leti@mail.ru)

**AGENDA / СОДЕРЖАНИЕ**

[***1.***](#_gjdgxs) ***BASIS FOR DEVELOPMENT 4***

[***2.***](#_35nkun2) ***НАЗНАЧЕНИЕ РАЗРАБОТКИ 4***

[***3.***](#_30j0zll) ***DETAIL REQUIREMENTS 4***

[***4.***](#_1fob9te) ***ВЫСОКОУРОВНЕВЫЙ ДИЗАЙН 4***

[***5.***](#_3znysh7) ***WORK BREAKDOWN STRUCTURE AND EFFORTS ESTIMATION 4***

[***6.***](#_2et92p0) ***KEY MILSTONES 4***

[***7.***](#_1ksv4uv) ***CHOSEN TECHNOLOGY AND DEVELOPMENT FRAMEWORK 5***

[***8.***](#_3dy6vkm) ***CONFIGURATION AND CHANGE MANAGEMENT 5***

[***9.***](#_1t3h5sf) ***PROJECT SPECIFIC PROCESS, QUALITY PLAN, METRICS 5***

[***10.***](#_4d34og8) ***PROJECT RISKS 5***

[***11.***](#_2s8eyo1) ***PROJECT SCHEDULE (GANT DIAGRAM) 5***

[***12.***](#_17dp8vu) ***TEST PLAN 5***

[***APPENDIX 1. TEST CASES 6***](#_44sinio)

[***APPENDIX 2. TEST RESULTS 7***](#_26in1rg)

[***APPENDIX 3. DEFECTS 8***](#_lnxbz9)

1. **BASIS FOR DEVELOPMENT**

Development is carried out for this project is by using VB language, it will be divided into four main steps including the interface, coding the ciphering method, design the database and final running, designing and reading the card.

Personal Information Encryption

PIE

1. **DEVELOPMENT GOALS**

The software project is intend to create an easier way, more secured and more convenient way of finding the saving personal information, such as files, names, ages, location.. ect of one or multiple people by giving each person a specific code that could be easily find just by scanning the code of that peron, each person has his own unique code that him and only him has it

1. **DETAIL REQUIREMENTS**

1 - The user shall be able to register in the database

2 - The user shall be able to login to his/her account

3 - Registered users shall be given a specific code to use for a login

4 - Users shall be able to save personal information such as names, ages, passports, personal information and so on

5- Each user shall have access only to his personal account and no one else

1. **HIGH LEVEL DESIGN**

1- Graphic user interface

2- Designing a database

3- Transferring users information (names, ages, heights, phones numbers.. ect)\*

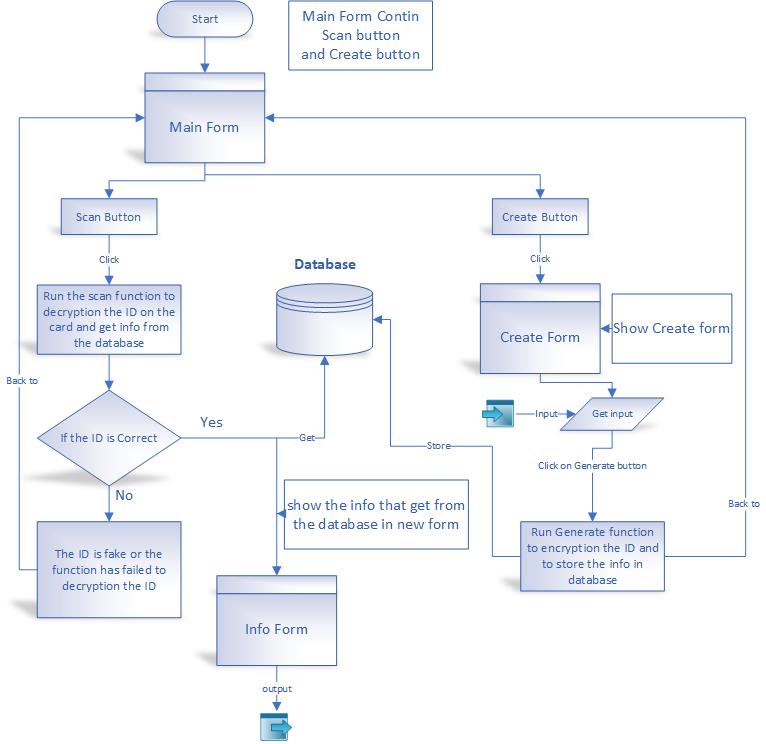
\*For now it only includes text information

4- Giving random numbers to each user after transferring the information to the database so that when the code is scanned and deciphered it would be easier for the program to find the meant user

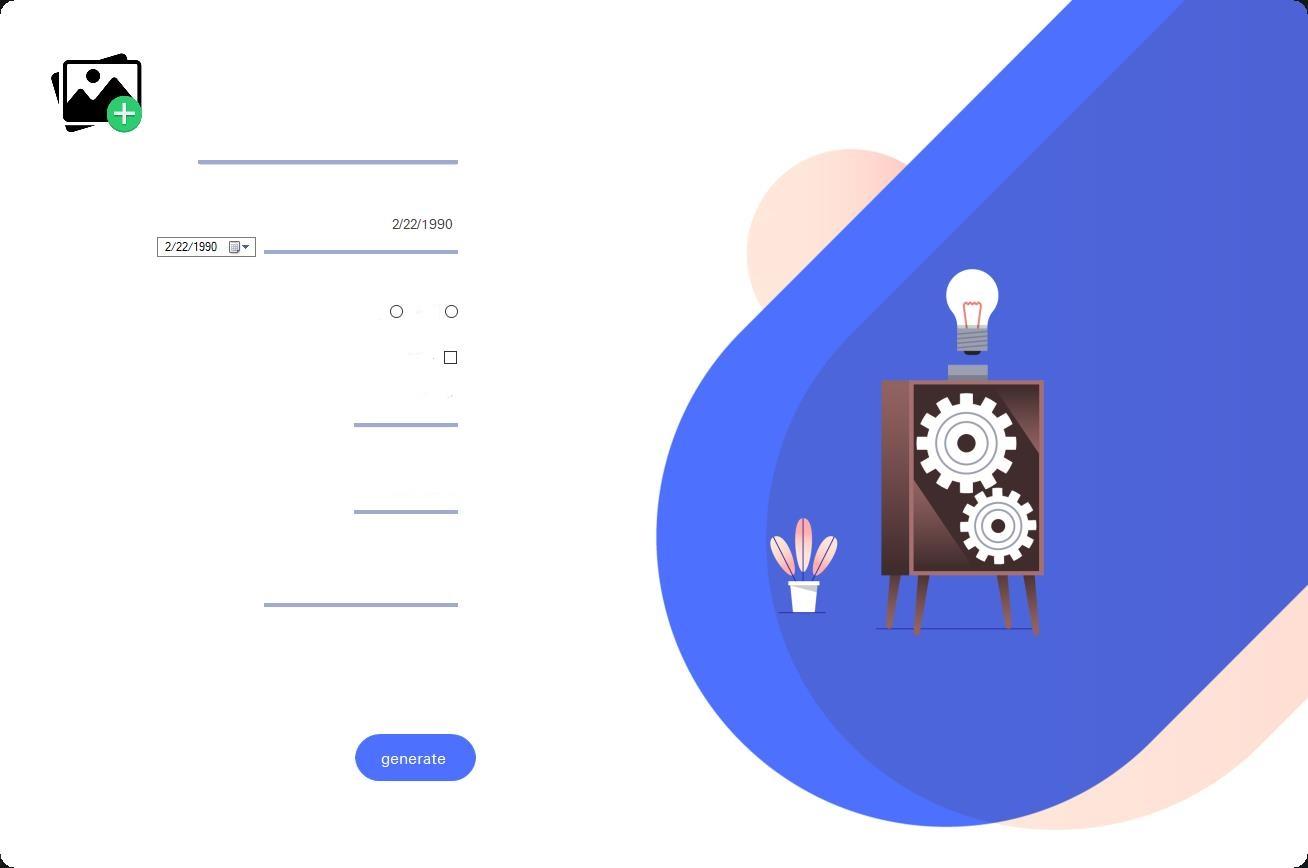
5- Designing a specific pattern for each user (code pattern)

7- Scanner platform for reading the codes

This is a simple data structure of the program and how it works



Here we shall be having some basic pictures of the program



#1 is to open a card that’s already been made.

#2 is to save a card after being ciphered

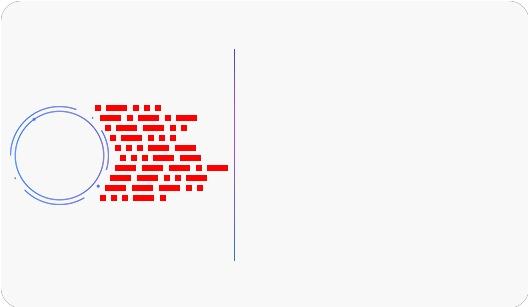
#3 is to exit or minimize the window

#4 is the card after being made

#5 is to create the card after putting all the information inside

#6 is to quick scan a card using the camera

And here how the final card will be looking like (This is not a final version of the card)



1. **WORK BREAKDOWN STRUCTURE AND EFFORTS ESTIMATION**

First of all comes the graphic interface designing by using the VB program that comes from Microsoft, it’s an easy part to design pictures and put them into the program, then comes the database part, then the method of encoding and decoding and the last part which is scanning and reading the code.

Let's start the major part which is the encoding and then we shall be talking about the graphic and the database

Summarizing the tasks that the program must do will have three main tasks:

1 - Encode code configuration

2. Decode analysis

3 - Query the database and fetch the correct information after decoding

Encoding:

The first part of the program is how to configure the code as a pattern and how to draw this style on the card. Personally, as I said earlier, I will quote from Morse code to create my code as I said instead. The point in the Morse code I draw a square and replace the line I draw a rectangle I will call these boxes. And the "Shapes" formats, so I will encode the ID in this picture according to a set ofthe rules :-

1 - Each ID number should consist of 9 classes, for example the following number (952823617.)

For example, each order contains 5 shapes

3 - Clean in the third dustbin of the ID key number on which we define the encryption pattern what this means

For example, if the third place represents the number two, the number 5 will be encoded in the form of a rectangle and then three squares and then a rectangle. If the third decimal of the ID represents the number 1, then the number 5 encoding will vary there will be, for example, four rectangles and then a square if the third rank will be the key on which we will be and decrypt.

4. ID will be a random number of 9 times and maximum number of IDs that can

The software configuration is 10 billion (which means 10 billion records).

After these rules have been developed, it is now time for programming. The first problem I have addressed is is how to tell the computer that for example, the number 5 represents a rectangle, three squares, then a rectangle? After thinking carefully I came up with the idea that Bali

The computer converts the normal number to the 0 and 1 binary system, which is also considered as a type of decryption the code is ASCII, so I was told to make the square equal to 1 and the rectangle equals 0 instead of saying the number 5 represents a rectangle and then three squares and then a rectangle I say that the number 5 is equal to "01110", where 1 is called a function that draws a square and 0 is called the function that draws a rectangle can easily handle the function and is easy to process so any code I just type with this "00001" will have 4 rectangles and one box will remember this is detailed in the decoding phase and the code steps are to configure the code

Decoding:

After successfully configuring the code, the next step is to learn how to make the program analyze and return the code to the ID number to inquire about in the database and to fetch the desired information and the configuration of the code is quite different for decoding it may be the most difficult and complex part is how to decode we will deal here with an image that contains symbols (code) and image is a matrix we will deal with a matrix and try to find the location of the code in this synonym and its analysis will be done through a set of steps

1. **KEY MILSTONES**

1- Develop the interface

2- Develop the database

3- Develop the coding

4-Devople the way of enciphering

5- Designing the pattern

6- Transferring patterns to numbers

7- Reading numbers and matching them to the database

8- Recall the information saved into the database to the interface

|  |  |  |  |
| --- | --- | --- | --- |
| **## / №** | **Start date** | **End date** | **Description** |
| 1 | 18.03.19 | 25.03.19 | SOW developed, reviewed and signed / |
| 2 | 25.03.19 | 01.04.19 | Designing the code |
| 3 | 01.04.19 | 08.04.19 | Designing the database and connect it to the program |
| 4 | 08.04.19 | 15.04.19 | Testing the coding of the enciphering method and its connection to the database |
| 5 | 15.04.19 | 22.04.19 | Designing the interface of the whole program |
| 6 | 22.04.19 | 29.04.19 | Attaching the code to the interface and testing the program |
| 7 | 29.04.19 | 06.05.19 | Coding the scanning way of reading the code |
| 8 | 06.05.19 | 13.05.19 | Course project completed, presentation and final report prepared for defense |

1. **CHOSEN TECHNOLOGY AND DEVELOPMENT FRAMEWORK**

The technologies and development framework was made by VB language coding using Windows WPF, the UI was created by using XMAL (Extensible Application Markup Language) and the program used to connect all these together Visual Studio

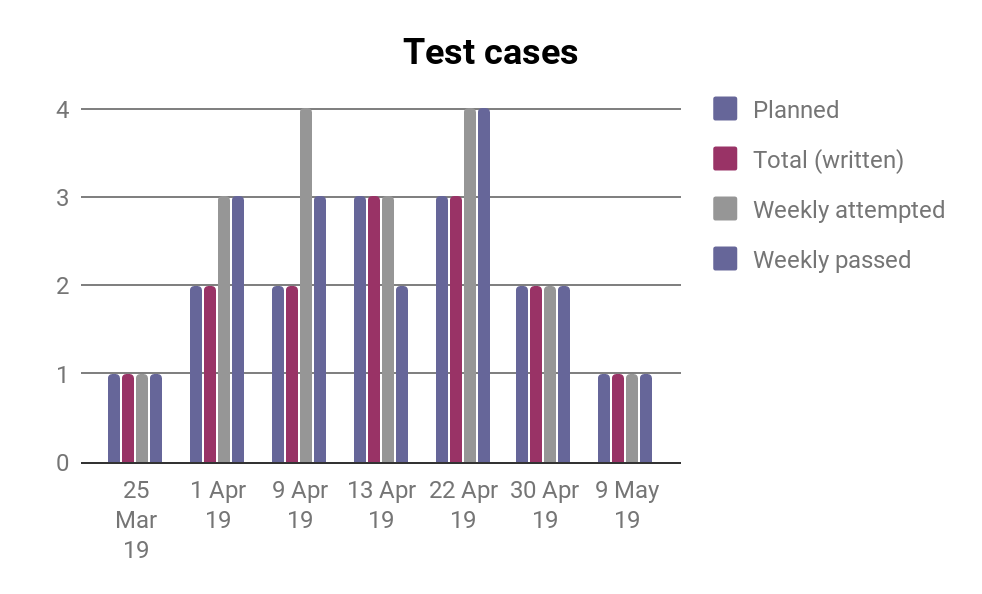
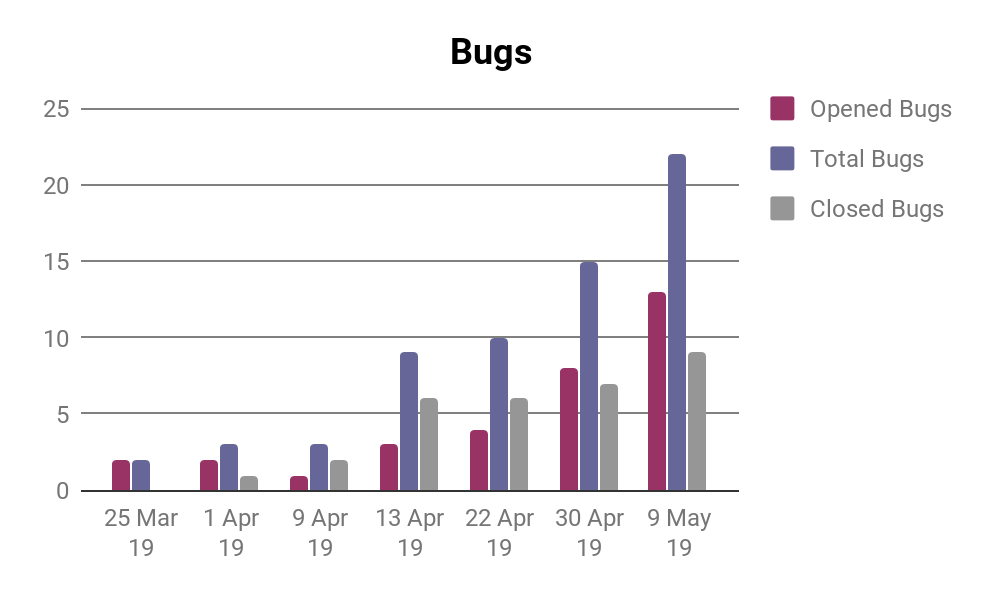
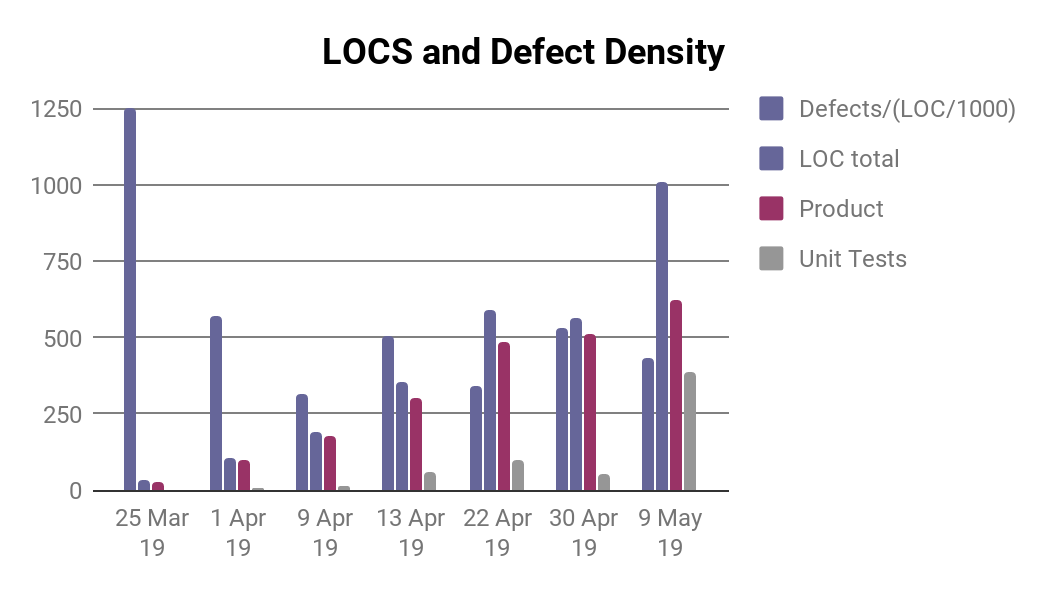
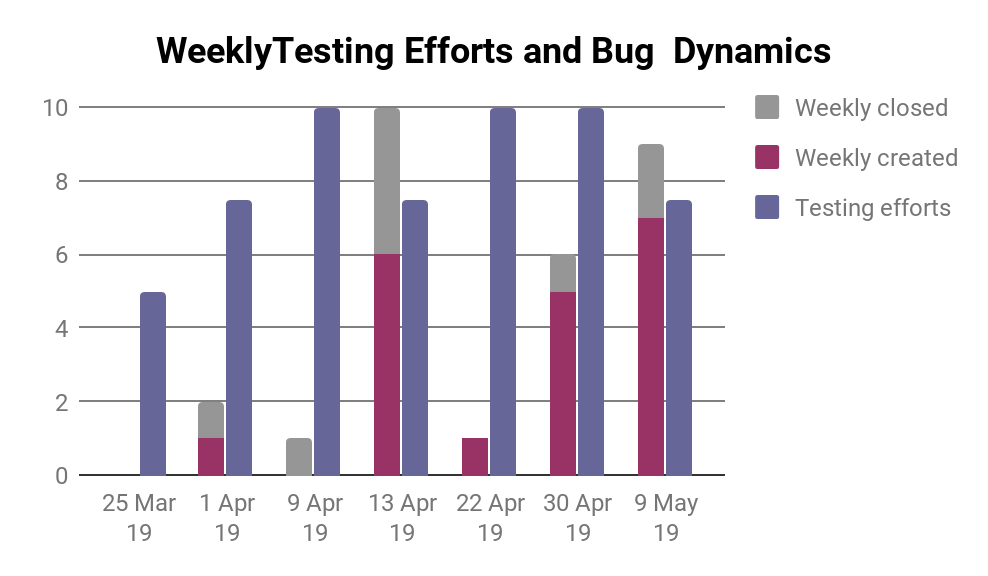
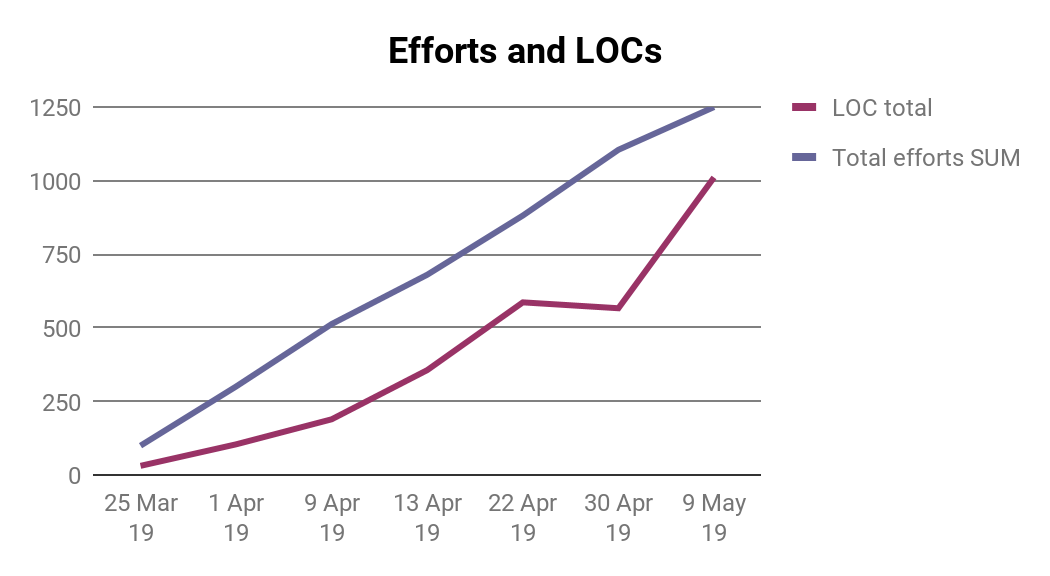
1. **CONFIGURATION AND CHANGE MANAGEMENT**

All the information are inside my Github

<https://github.com/asibra/Card-Encoding>

1. **PROJECT SPECIFIC PROCESS, QUALITY PLAN, METRICS**

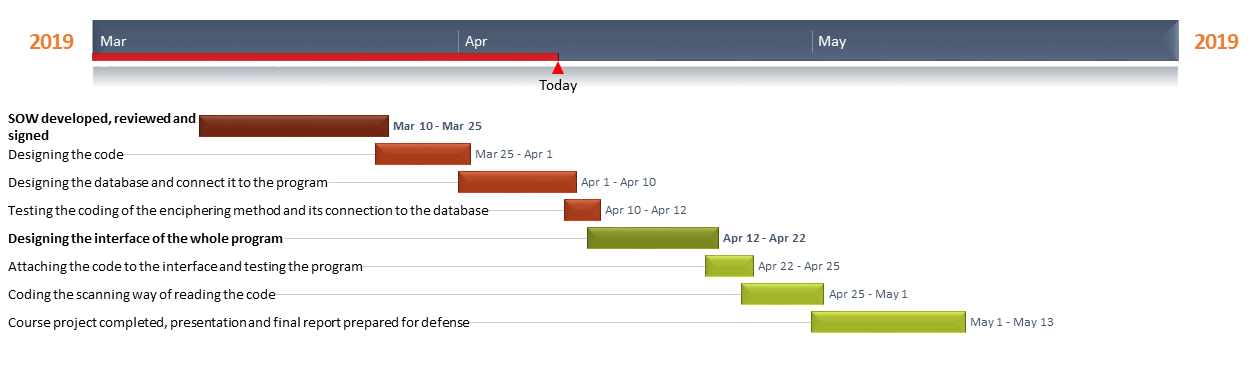
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Project Name: Personal Information Encryption** |  |  |  |  |  |  |  |
|  | **Project Manager: Salwan Luay Chasib** |  |  |  |  |  |  |  |
|  | **Test Lead: Salwan Luay chasib** |  |  |  |  |  |  |  |
|  | **Report Date** | **25 Mar 19** | **1 Apr 19** | **9 Apr 19** | **13 Apr 19** | **22 Apr 19** | **30 Apr 19** | **9 May 19** |
|  | **Project week number** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | **Team efforts (work-hours)** |  |  |  |  |  |  |  |
| c | **Total weekly efforts** | 9 | 18 | 19 | 15 | 18 | 20 | 13 |
|  | Management & support | 2 | 6 | 5 | 4 | 0 | 5 | 3 |
|  | Development efforts | 5 | 9 | 10 | 8 | 14 | 11 | 7 |
|  | Testing efforts | 2 | 3 | 4 | 3 | 4 | 4 | 3 |
| c | **Total efforts SUM** | 9 | 27 | 46 | 61 | 79 | 99 | 112 |
|  | **Lines of code** |  |  |  |  |  |  |  |
| c | **LOC total** | 32 | 105 | 190 | 357 | 587 | 567 | 1011 |
|  | Product | 30 | 100 | 175 | 300 | 488 | 512 | 622 |
|  | Unit Tests | 2 | 5 | 15 | 57 | 99 | 55 | 389 |
| c | **Defects/(LOC/1000)** | 62.5 | 28.6 | 15.8 | 25.2 | 17.0 | 26.5 | 21.8 |
|  | **Bugs (numbers)** |  |  |  |  |  |  |  |
|  | **Total Bugs** | **2** | **3** | **3** | **9** | **10** | **15** | **22** |
|  | Opened Bugs | 2 | 2 | 1 | 3 | 4 | 8 | 13 |
| c | Closed Bugs | 0 | 1 | 2 | 6 | 6 | 7 | 9 |
|  | **Weekly bug dynamics (numbers)** |  |  |  |  |  |  |  |
| c | Weekly created | n/a | 1 | 0 | 6 | 1 | 5 | 7 |
| c | Weekly closed | n/a | 1 | 1 | 4 | 0 | 1 | 2 |
|  | **Test cases (numbers)** |  |  |  |  |  |  |  |
|  | **Planned** | 1 | 2 | 2 | 3 | 3 | 2 | 1 |
|  | **Total (written)** | 1 | 2 | 2 | 3 | 3 | 2 | 1 |
|  | Weekly attempted | 1 | 3 | 4 | 3 | 4 | 2 | 1 |
|  | Weekly passed | 1 | 3 | 3 | 2 | 4 | 2 | 1 |
| c | Planned - Total | 0 | 0 | 25 | 0 | 0 | 0 | 0 |
| c | Weekly Attempted/Total (%) | 100% | 150% | 25 | 100% | 133% | 100% | 100% |



1. **PROJECT RISKS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Risk No.** | **Risk Name** | **Probability (from 0 to 9)** | **Impact** | **Rate** | **Mitigation plan** |
| **25.03.19** | **1** | **When the code pattern was designed, the dots and the lines were overlapped between each other** | **3** | **M** | **M** | **1. An increase in the amount of x when the following figure (ie, the next figure is shifted horizontally from the first by a certain amount to separate the two shapes) and an increase in the amount of Y (for the horizontal offset by a certain amount)** |
| **26.03.19** | **2** | **Isolate the code for later reading** | **3** | **M** | **M** | **1. Configure the code in one color, which is the red color for the purpose of isolating it, ie, when the scan of the card or image is done, all but the red colors are converted to black color.** |
| **27.03.19** | **3** | **Number of shapes that are encoded in a single number** | **5** | **M** | **M** | **Each number is encrypted into only two forms, ie, a square and a rectangle. We can only encode two digits. If we make each number encoded into three forms, we can only encode 8 numbers. The number 2 encryption would be. . \_ The three digit encryption will be. . . Encoding 4 will be \_. . And encryption 5 will be \_ \_. Encoding 6 will be \_ \_ \_ and encryption will be \_. \_ The encryption will be \_. \_ These are all the possibilities that can be configured so you encrypt each number to 5 forms to increase the amount of possibilities available and make the code more secure** |
| **28.03.19** | **4** | **Number of ID mattresses that encrypts** | **3** | **M** | **M** | **1. Make the number of mattresses 9**  **2. Maximum number of ID is 999999999** |
| **29.03.19** | **5** | **Make the code difficult to increase security** | **3** | **M** | **M** | **1.Configure more than one encryption pattern)** |
| **06.04.19** | **6** | **Link the database with the program** | **2** | **M** | **M** | **Use the code: Dim con As New OledbConnection("Provider=microsoft.Jet.oledb.4.0DataSource=D:\mydata.mdb;")** |

1. **PROJECT SCHEDULE (GANT DIAGRAM)**



1. **TEST PLAN**

|  |  |
| --- | --- |
| **Test type** | **Test case description** |
| 1. Alpha Testing | identify all possible issues or bugs before releasing the application |
| 2. Test Scope | The components of the system to be tested (software, middleware, buttons,etc ) |
| 3. Example Testing | By real-time testing each step of the program during and after finishing the coding |
| 4. Functional Testing | Checks only on the output to find if it is as per the requirement or not (Black-box testing) |
| 5. Security Testing | Check how the software is secure from internal and external threats. This testing includes how much software is secure from the malicious program, viruses and how secure and strong the authorization and authentication processes are. |

**APPENDIX 1. TEST CASES**

|  |  |  |
| --- | --- | --- |
| **Test type** | **Test case ##** | **Test case description** |
| 1. The efficiency of the code algorithm test | 1.1 | Measuring the speed of making the code when pressing the coding button in the application and the results achieved is less than one second |
| 2. Algorithm accuracy test | 2.1 | Test the accuracy of the code-making algorithm so when encoding a certain number and then manually decode the code we see if the resulting number matches the written number |
| 3. Print and scan code test | 3.1 | Pring the code as an image and then scanning it inside the program |
| 4. Database stability test | 4.1 | Test the database for how to add records as information to the database and check if the information is saved in the database or not |
| 5. Information retrieval test | 5.1 | Check the retrieval of information from the database where we see the store ID and see if the information is fetched for this ID by Datagridview |

**APPENDIX 2. TEST RESULTS**

|  |  |
| --- | --- |
| **Test type** | **Results**  **Attempted / Passed successfully** |
| The efficiency of the code algorithm test | Passed |
| Algorithm accuracy test | Passed |
| Print and scan code test | Failed |
| Database stability test | Passed |
| Information retrieval test | Passed |

This is test report on 14.05.2018

**APPENDIX 3. DEFECTS**

**List of defects found**

|  |  |  |
| --- | --- | --- |
| **##** | **Description** | **Current state (found, fixed, verified, assigned, open)** |
| 1 | The pattern of the code can only be read from a good quality picture and it can not be scanned from a picture captured by a camera | open |
| 2 | The pattern of the code was crossed between each other so the dashes and the dots of the pattern was looking like a one drawing | fixed |
| 3 | One method of encrypting the information was easy to be cracked by thinking about the code for more than 15 mins | fixed |

**Defect statistic on categories**

|  |  |
| --- | --- |
| **Defect categories** | **Number**  **Total / Open** |
| Critical | 0 / 0 |
| Major | 0 / 0 |
| Regular | 1 / 1 |
| Minor | 0 / 0 |
| Enhancements | 0 / 0 |

Defect statistic on state

|  |  |
| --- | --- |
| **Defect states / Количество дефектов по состояниям** | **Number** |
| **Found** | 0 |
| **Fixed** | 1 |
| **Verified** | 0 |
| **Assigned** | 0 |
| **Open** | 1 |

Defect report on 14.05.2018