

**College of Computer & Info Sciences Department of Software Engineering**

Hidden Message.

#### Case study: message protection Mobile application to protect/secure messages.

Assignment #2

|  |  |  |
| --- | --- | --- |
| # | Student Name | ID |
| 1 | Hussa AlQuraishi | 437200828 |
| 2 | Fouz AlRabai | 437200482 |

**Table of content**

[Introduction 3](#_Toc33996255)

[Source Code 3](#_Toc33996256)

[Execution results 4](#_Toc33996257)

[References 6](#_Toc33996258)

# Introduction

In this assignment, it was assigned to us to implement a cryptosystem. We implement cipher application by using Keyword columnar algorithm that operates in Java programing language.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S | P | Y | M | A | N |
| 5 | 4 | 6 | 2 | 1 | 3 |
| A | t | t | a | c | k |
| A | t | D | a | w | n |

The Columnar Transposition Cipher is a form of transposition cipher just like Rail Fence Cipher. Columnar Transposition involves writing the plaintext out in rows and then reading the ciphertext off in columns one by one [1] . First of all, we need to create matrix of size n\*m, (n= rows number , m=columns numbers) where m equals to numbers of keyword's characters and n equals to plaintext divided by columns number. In the first row write the key down ,after that in the second row write 1,2,3,… defined by the alphabetical order of the letters in the keyword, then write your plaintext. Finally, read off in columns, in the order specified by the keyword. For example , the plaintext: AttackAtDawn , keyword: spyman.

# Source Code

# Execution results

# References

|  |  |
| --- | --- |
| [1] | GeeksforGeeks, "Columnar Transposition Cipher," GeeksforGeeks, 2020. [Online]. Available: https://www.geeksforgeeks.org/columnar-transposition-cipher/. [Accessed 1 March 2020]. |

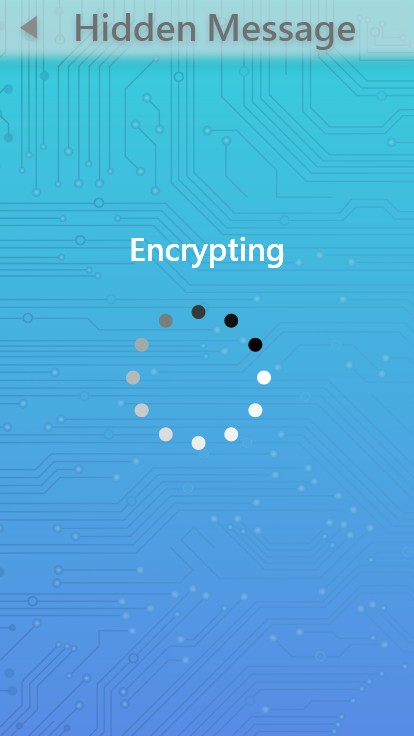
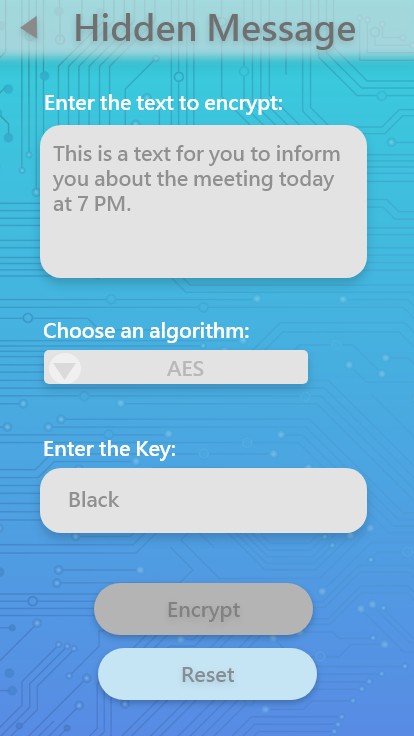
## User Interface

First of all Hidden message is a mobile application for decrypting or encrypting messages among different people, for such in-complex system the interface is simple and conforms with the principles of good interface design. Users of Hidden Message are mostly non- technical people, hence the straightforward user interfaces. In order to avoid break-downs in interaction, we must ensure that UIs do not contain elements that may possibly prevent users from using Hidden Message system or create an unnecessary obstacles to them.

Bearing in mind the reason for every principle of good interface design, We ensured that Hidden Message system can be used by a wide range of users of different backgrounds. The main functions of this application are well-placed; hence, a high degree of visibility has been achieved in the design of Hidden Message. Additionally, the application obtains the graphical icons that are designed according to the principle of affordance. It is also consistent both aesthetically and functionally.

High Visibility is met. After each operation that has been made E.g. encrypting a message the system will display a text showing that the message is processing with a dotted circle, this will assure to users that the action taken has been accomplished, which in this case is the message is being processed currently, in other words the system’s status is visible to the user (see figure 7.3.2 ), thus users will never guess about what action they have taken and the consequence of doing so.

Hidden Message provides Feedback to its users, Another example is when a button is pressed the color will turn to grey, this will indicates to the user that this button is pressed. In user’s perspective, it is obvious when the system function has been triggered (see figure 7.3.1).

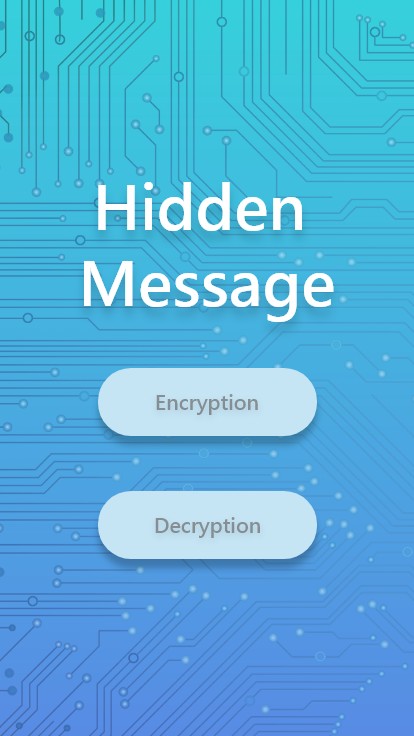


*Figure 7.3.1: encrypting interface a button is pressed after filling all fields*

*Figure 7.3.2: after pressing Encrypt button the processing interfaces will be displayed*

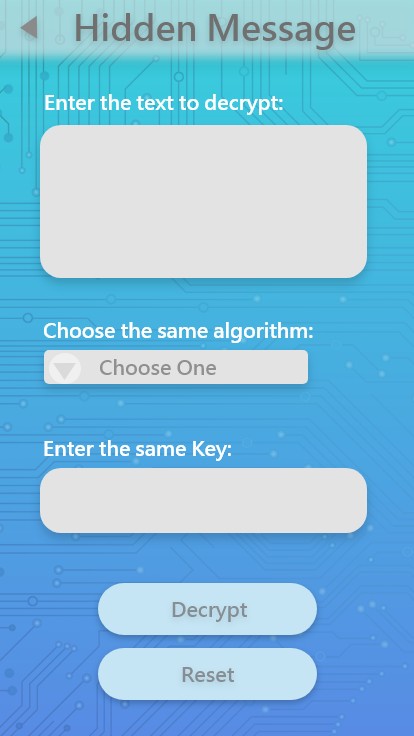
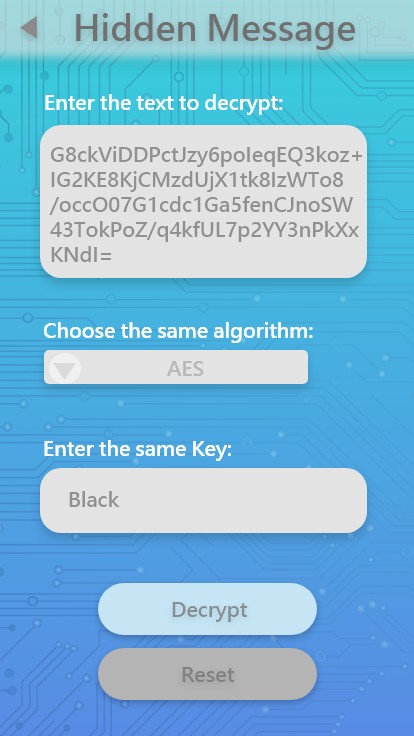
The interface of Hidden Message affords high usability and it includes Learnability, Efficiency, Satisfaction for its users. Learnability is met in this system by having straightforward user interface designs that are predictable in layout and navigation. Not having interfaces that have crowded with too much information also benefits learnability. Moreover elements such as buttons are repeated in a rhythmic pattern so that user expect to find while using the software which makes the users learn easily.

Affordance of Hidden Message system is provided by having simple button shapes users will use these buttons to perform the wanted tasks. Each button has clear text that indicates understandable meanings, these clear text will give users a clue on how to perform the tasks *(see figure 7.3.3)*. It’s clearly known for users that the 1st button is for Encryption a message.



*Figure 7.3.3, home page of Hidden Message*

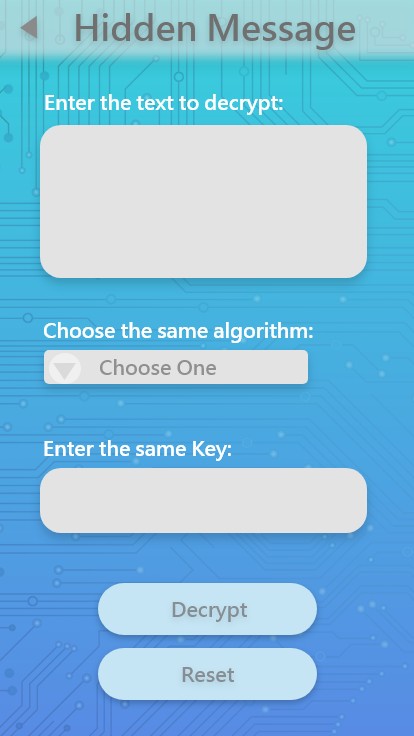
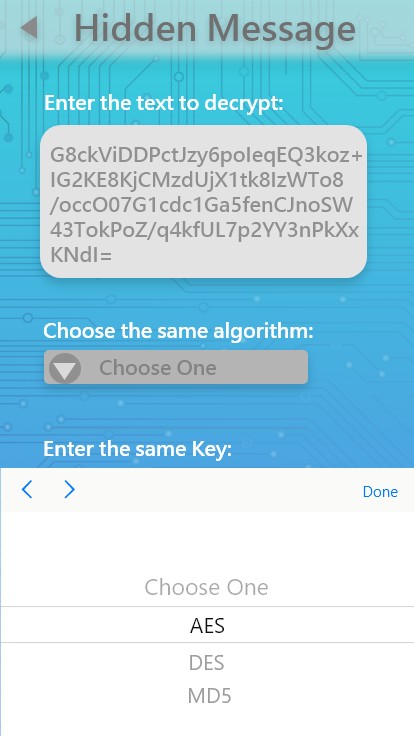
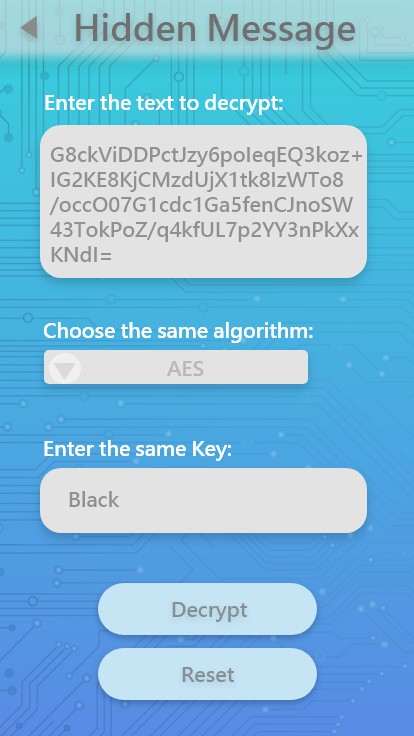
Learnability example, Re-setting all filled fields to empty fields can be done easily by pressing the reset button *(see figures 7.3.4 , 7.3.5)* of that process.



*Figure7.3.4: a filled fields and a pressed reset button. Figure7.3.5: all fields are empty.*

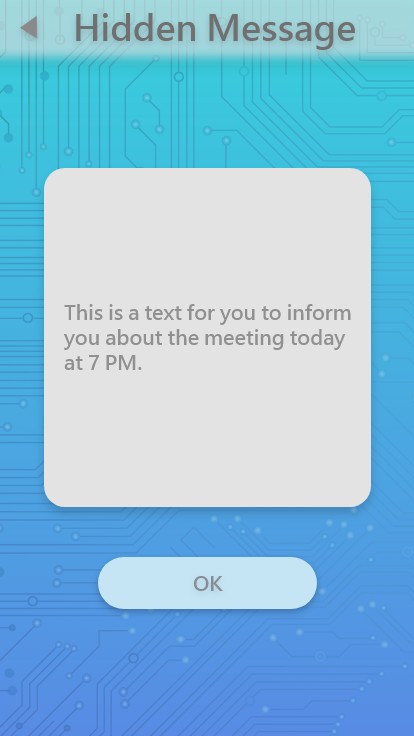
Efficiency is also improved since each interface is familiar and not complicated, new users will learn how to perform tasks such as encrypting a message immediately, thus users who have no experience in encryption and decryption will still be able to decrypt/encrypt their messages easily.

Aesthetic backgrounds and round shapes with modern fonts increase the satisfaction and enjoyment of users when using the system. *(see figures 7.3.6-10)* for decryption message process which shows the high usability of Hidden Message.

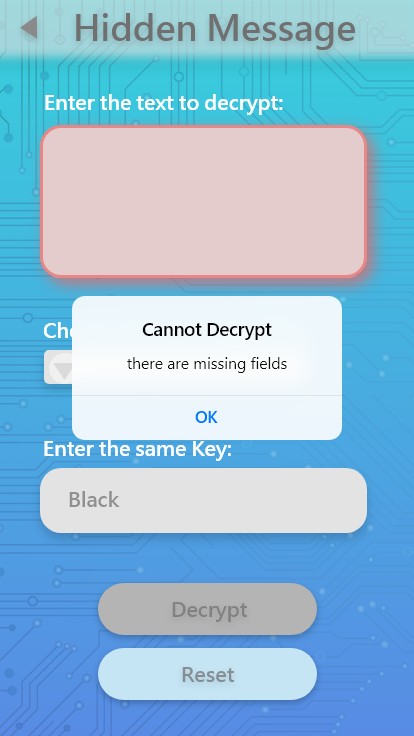
*Figure 7.3.6: after choosing decryption button Interface with An empty fields.*

*Figure 7.3.7: filling fields with needed information Figure 7.3.8: filled fields.*

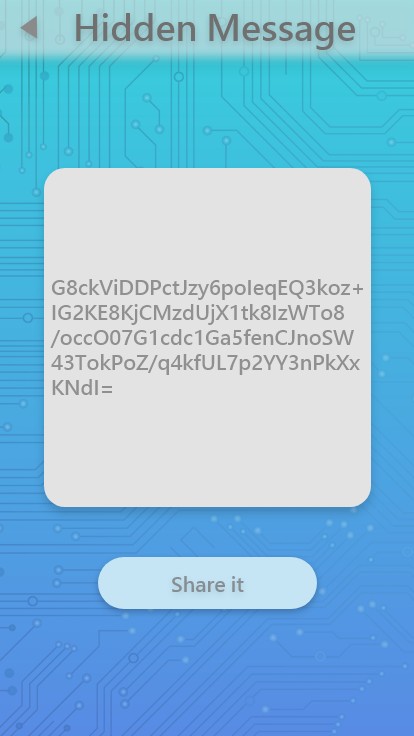
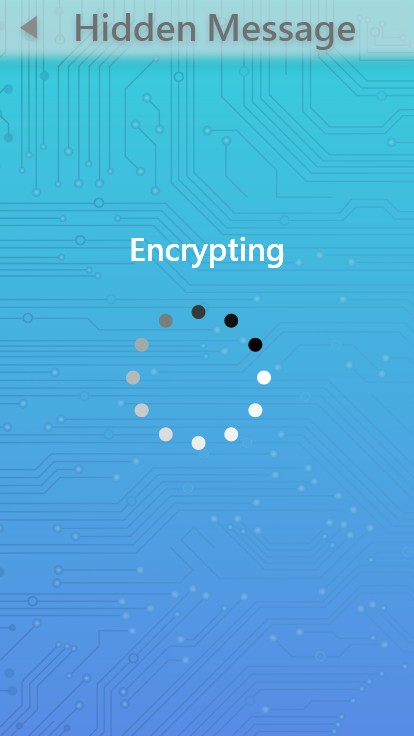
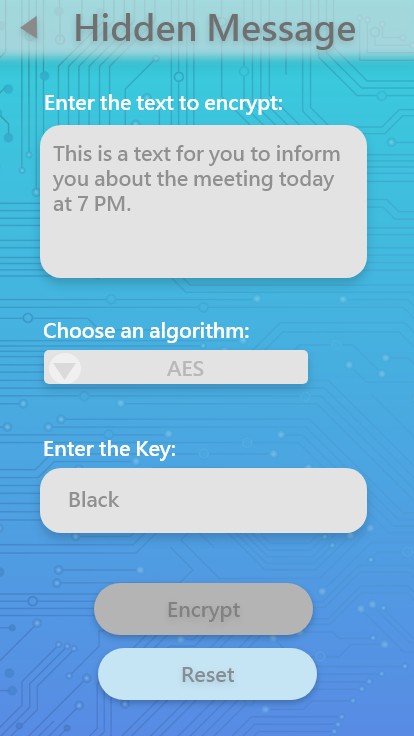
*Figure7.3.9: decrypting the cipher text. Figure7.3.10: a plain text has appeared to user*

Errors are recoverable in Hidden Message system for instance, a user may press encrypt/ decrypt button without filling all fields. A red border will appear the missing field/s as well as the field will turn to a faded red color. a message for error indicating will also appear. *(See figure 7.3.11)*



*Figure 7.3,11: when a button is pressed but there is/ are some empty fields*

For encryption process.



*Figure 7.3.11: Encrypt button is pressed. Figure 7.3.12: Encrypting the plaintext. Figure 7.3.13: Cipher text is appearing to the user*

# Challenges

There are several challenges that we may encounter, the main challenge is the Text message protection has a lot of methods/algorithm and we’re unfamiliar with it.

Also, the strength of the cryptography algorithm depends on the secrecy of the key, if the secret key between the sender or receiver is out publicly, then the security of the message decreases to a low level and can be decrypted easily.

Other group work-related challenges:

1. other projects may slow down our work on this project.
2. Group Meetings, since we have different courses, therefore figuring out a free time for all of us might be a bit challenging.
3. Choosing the appropriate architecture is gained by experience.

# Projection

1. Understand the proposed software design methods and styles.
2. Analyze and design software systems and apply the suitable architecture styles.
3. Learning about how to create and choose the most suitable design for the solution by using design patterns.
4. To learn more about security types and get general ideas about this subject.
5. document and deliver the final and correct product on time.
6. Decision making.
7. Problem solving.
8. Attention to details.
9. Improve quality and functionality offered by the system.