
Software Requirements Specification

for

CryptoKnight

Version 1.0 approved

Prepared by

**Akshat Garg
(150911066)**

**Shraddha Duseja
(150911100)**

**Aditi Krishna
(150911118)**

9th March, 2017

Table of Contents

1. Introduction.....	1
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope	1
1.4 References.....	2
2. Overall Description.....	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints.....	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	4
3. External Interface Requirements	4
3.1 User Interfaces	4
3.2 Hardware Interfaces	4
3.3 Software Interfaces	5
3.4 Communications Interfaces	5
4. System Features.....	5
4.1 Brute-Force attacker	5
4.2 Dictionary attacker.....	6
4.3 Password strength checker.....	7
5. Other Nonfunctional Requirements.....	8
5.1 Performance Requirements.....	9
5.2 Safety Requirements	9
5.3 Security Requirements	9
5.4 Software Quality Attributes.....	10
6. Legal Requirements.....	11
Appendix A: Glossary.....	11
Appendix B: Analysis Models.....	12

1. Introduction

1.1 Purpose

The aim of the project is to create a software that will act as a penetration testing suite for cracking login modules. The software would be able to crack passwords and if required usernames. The software would be able to recommend strong passwords to users and test the strength of passwords provided by the users. The software will act like a medium for professionals to test their modules for weaknesses, and to calculate the various login parameters that provide the optimum balance between ease of use and security.

1.2 Intended Audience and Reading Suggestions

The software will help the testers, the network specialists and also the security consultants for vulnerability testing in order to identify potential problems, along with the solutions to overcome the same. Our software consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.

The various brute force attacks that our project will mainly focus on are listed below:

- Dictionary attack: A dictionary file can be tuned and compiled to cover words probably used by the owner of the account that a malicious user is going to attack.
- Search attack: Search attacks will try to cover all possible combinations of a given character set and a given password length range.
- Rule-based search attack: To increase the combination space coverage without slowing too much of the process, it's suggested to create good rules to generate candidates.

In order to read the document, the testers should basically focus on the Product Scope, Product Functions and System Features. The security consultants should review design and implementation of constraints and the security requirements. The network specialists should go through the sections that describe communication interfaces, operating environment, user classes and characteristics.

1.3 Product Scope

Explicit mentioning of scope can be the penetration testing of login modules especially for databases, websites and other systems where high level of security is desired. The software is designed to aid the pen tester find flaws in their clients systems and take appropriate action. Also in certain cases the software can be used to crack passwords that the user might have forgotten. Brute force attacks work by calculating every possible combination that could make up a password and

testing it to see if it is the correct password. A dictionary attack is similar and tries words in a dictionary or a list of common passwords instead of all possible passwords.

1.4 References

IEEE 830-1998 Standard for writing software requirement specification document.

2. Overall Description

2.1 Product Perspective

It is a new, self-contained product which will act as a penetrating testing suite for cracking various login modules. The software would be able to crack or decipher passwords as well as usernames, if required. It is an authorized simulated attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data. The software would also be able to recommend strong passwords to the user and test the strength of the passwords provided by the user to enable and enhance security within the login modules, making it difficult for any outside entity to crack into the system.

2.2 Product Functions

The process encompasses the identification of target systems and a specific goal, it then reviews the available information and data. It then takes up various means to attain that goal. It can help in determining the vulnerability of the system to such attacks. The product will incorporate the following functions:

- It should be able to conduct brute force attacks within the specified boundaries.
- It should be able to conduct dictionary attacks using the given text file of the dictionary.
- It should use multithreading to implement the different functionalities.
- It should calculate the strength of the password provided.
- It should generate random strong passwords.

2.3 User Classes and Characteristics

The product can be used by users of different communities and backgrounds. The implementation of the product will be variable, based upon the type of need of the user. A few of the user classes are as follows:

- Naïve users- These users limit their operations to the basic functionalities of the software which includes the determination of the strength of password and the appropriate username that would enable them to support a secure system, which cannot easily be cracked into.
- System administrators- These type of users explore the advanced functionalities of the product which includes the implementation of brute force attack and dictionary attack that helps them crack into a login module. The other uses are to modify and optimize their security constraints, for example limiting the number of characters in the password and the minimal requirement as to what the password string must consist of.
- System owners- Another very important use of the product is to keep a check on the vulnerability of the system, which can be sheltered with the help of employing methods which engages in putting some constraints to the password to make the system more secure and reliable.

2.4 Operating Environment

The software should be supported by operating systems such as Windows with versions 8, 8.1 and 10, respectively, to enable the naïve users as well as the specialized users to utilize the software at the simplest level.

2.5 Design and Implementation Constraints

The specific constraints bounding the design constraints of the said software are:

- Language to be used : JAVA
- Hardware constraints: The system is limited in hardware resources, with specific issues being the lack of a discrete Graphics Card and the limited number of threads that can be run by the target system.
- The software cannot be allowed to store any of the passwords it cracks, due to security and legal concerns. The system hence cannot use any Databases.
- Use of any existing tools and software, closed source or open source alike should be prevented altogether to minimize the risk of any dependencies and legal issues.
- The developers will be responsible for all sort of maintenance and hence the design is driven with this in mind.

2.6 User Documentation

The user documentation will be delivered along with the software, which includes online help, video tutorials and user manuals which would support the user and help them achieve a better understanding of the software and enable them to incorporate security into their systems easily and

efficiently. The user manual should be available in the Open Encoded Format such as HTML, XML for the specialized users of the community, which is accessible with all current web technologies. It should also be available in Hybrid Format such as PDFs, as this format is used by all kind of users, including the naïve users.

2.7 Assumptions and Dependencies

Under budget and time constraints are common techniques that discover vulnerabilities. It aims to get an unhandled error through random input. Errors are useful because they expose more information generally. The user uses random inputs to access less often used code paths. This is the concept of brute force attack which is basically a trial and error method used to decode encrypted data such as passwords. As in the case of dictionary attack, the code path will be pre-given. In this type of penetration attack, attempts are made gain access to a computer system by using a very large set of words to generate potential passwords. The source code of the file containing all the possible combination will be given before-hand.

3. External Interface Requirements

3.1 User Interfaces

All the user interfaces for the system will follow be designed in accordance to the 8 commonly accepted characteristics of a well-designed user interfaces, which are:

- Clear
- Concise
- Familiar
- Responsive
- Consistent
- Attractive
- Efficient
- Forgiving

No specific style guides will be followed as the emphasis of the software is on the algorithm which will perform the actual operations. The nature of the operations of the software entails the lack of requirement of any keyboard shortcuts for the operations of the software. The error messages should be concise and well detailed, informing the user in complete detail what the cause of the error is. Since the target of the software is professionals who might prefer to modify the source code of the

above software to suit their professional needs and unique requirements, the need of detailed error messages is self-explanatory.

3.2 Hardware Interfaces

The device supported by the product are fully functional computers (conventional desktops or laptops).

3.3 Software Interfaces

The software being produced is intended to act as a self-contained system, minimizing all dependencies on softwares and libraries outside the control of the developers. The product hence does not connect to any other specific software components apart from the operating system of the system the software is installed and operated on. The target operating system for the product is Microsoft Windows 8. The product requires no services from external sources.

3.4 Communications Interfaces

The sole communication required by the software to external interface is that required for linking it to the login module to the software. However this does not require any particular communication functions.

4. System Features

4.1 Brute force attacker

4.1.1 Description and Priority

This feature allows the user to conduct brute force attack on a particular given login modules testing out all the possible combinations of passwords, within the specified constraints.

Priority: High

4.1.2 Stimulus/Response Sequences

- User opens the software and selects brute force as the option.
- The form for getting data from the user is generated.

- The user enter the parameters for the password, including the minimum and maximum length of the password and all the allowed characters for passwords.
- User selects if brute force is to be performed on the password field or the username field.
- The user then links the form where the attack is to be conducted and selects the field for trial and links the button for trial.
- The system then starts trying all possible combinations on the selected field, using the specified constraints till it either exhausts all possible combinations or it successfully manages to gain access.
- The system then displays a dialog with essential information including time taken and the correct password.

4.1.3 Functional Requirements

REQ-1: A login module which can be linked and cracked.

REQ-2: In case of inconsistencies in constraints, the user must be prompted with a error message and should re-enter the constraints.

REQ-3: In case of failure to login into the linked login module, the system must prompt the user to re-enter the constraints.

REQ-4: In case the login module throws any exception, due to the constraints or due to the number of attempts, the software must exit immediately, providing all the details, including the number of tries and the error provided by the login module.

4.2 Dictionary Attacker

4.2.1 Description and Priority

This feature allows the user to crack a particular login module by using a selected dictionary file, which is a text file having all the passwords which will be tested in the login module to crack the particular instance. This feature is one of the central features of the software and hence, the priority is high.

4.2.2 Stimulus/Response Sequence

- User opens the software and selects dictionary attack as the option.
- The form for getting data from the user is generated.
- The user enter the parameters for the password, including the minimum and maximum length of the password and all the allowed characters for passwords.

- User selects if dictionary attack is to be performed on the password field or the username field.
- The user then links the form where the attack is to be conducted and selects the field for trial and links the button for trial.
- The system then starts trying all passwords from the dictionary on the selected field, using the specified constraints till it either exhausts all possible combinations or it successfully manages to gain access.
- The system then displays a dialog with essential information including time taken and the correct password.
- Information including time taken and the correct password are displayed to the user.

4.2.3 Functional Requirements

REQ-1: The feature requires a linked login module.

REQ-2: In case the login module gives an error regarding the number of attempts, the software must quit immediately, providing the user with an error prompt specifying exactly the number of tries that were allowed before the error message.

REQ-3: In case the user enters wrong constraints, which are not accepted by the system, the software must exit with an error message informing of an inconsistency in constraints, which the user will be prompted to re-enter.

REQ-4: In case the linked dictionary is not a valid dictionary file for the software, the software must prompt the user regarding the same requesting a new dictionary before the attack actually begins.

REQ-5: If the dictionary is exhausted before the login module is cracked, the software must provide the user with the appropriate error message and suggestions for the next dictionary attack on the same module.

4.3 Password Strength Checker

4.3.1 Description and Priority

This feature allows the user to enter their password and then receive a score of their password strength based on an algorithm and the score must contain a numeric value as well as state the time it will be taken by the software to crack the system. This feature allows the user to make informed

decisions regarding their password choices and ensure higher security in their personal accounts. This feature is secondary to the software and hence the priority for the same is LOW.

4.3.2 Stimulus/Response Sequence

- The user selects the option to check password strength.
- The user is prompted to enter the password to be checked.
- The user enters the password they desire to check the strength of.
- The system calculates a score for the entered password.
- The system then displays the score achieved by the entered password and the user is provided with suggestions to improve the score if the score is below a certain minimum.

4.3.3 Functional Requirements

REQ-1: A uniform method for grading a password and calculating the time required to crack it is required.

REQ-2: In case the user entered password scores below a basic minimum, the user must be informed of the weakness of the password and provided with suggestions to improve the same.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software is required to have satisfactory performance, which can be reasonably expected from the specified hardware platform for both the dictionary attack and the brute force attacks. However as a general performance benchmark, the system should successfully crack all passwords under the length of 8 characters under a time period of 2 days (48 hours). This limit is considerably high as compared to other systems, as the design is constrained to processors and cannot use GPUs or any other form of cluster computing for the same. For the password strength predictor, the system should return the strength of the entered password in terms of the number of days it would take to brute force it under the time period of 3 seconds to be reasonably satisfactorily.

5.2 Safety Requirements

Due to the nature of the product and its possible malicious usage, most of the on-line services on the Web use technologies to protect the systems and other data from being vulnerable to the attacks being used here. Hence, to avoid loss of any form of privileges or perusal of criminal action, the users are requested not to use the software at any location, physical or virtual without prior permission from the concerned owners or their representatives. All owners wishing to test the security of their systems using this software are recommended to ensure isolation, using sandboxing or other technologies available at their disposal to ensure safekeeping of their data. Responsibility for ensuring safety of data and systems lies solely with the user and the developers will in way be held accountable in case of any loss of data.

5.3 Security Requirements

Any data entered by the user is not stored on any location, including but not limited to the system where the software is installed or remotely on developer's cloud. The cracked passwords, the dictionary and the passwords whose strength is being calculated is not stored at any location.

Due to the particular nature of the product and the lack of universality in Information Security laws globally, users are required to refer to the information security and associated laws in their country before using the software. The developers will not be held accountable to any crimes where the software might be directly or indirectly involved. Ensuring the legality of the software in a particular case lies completely on the users themselves.

5.4 Software Quality Attributes

The software is primarily aimed at professional penetration testers and considering the targeted use base, ease of use has been emphasized over ease of learning to provide better operational usage over long periods. The software is aimed to be maintainable and scalable, and the entire process of design and development has been directed by the same philosophy. The software will be portable, easily installed on various systems with different configurations. Due to the critical nature of the software, correctness has been laid strict emphasis on. All corner cases and extreme cases must be accommodated for. The system must be reliable producing the desired results whenever used under the specified conditions.

6. Graphical User Interface

The GUI for our project will be integrated with the help of the following GUI's listed below.



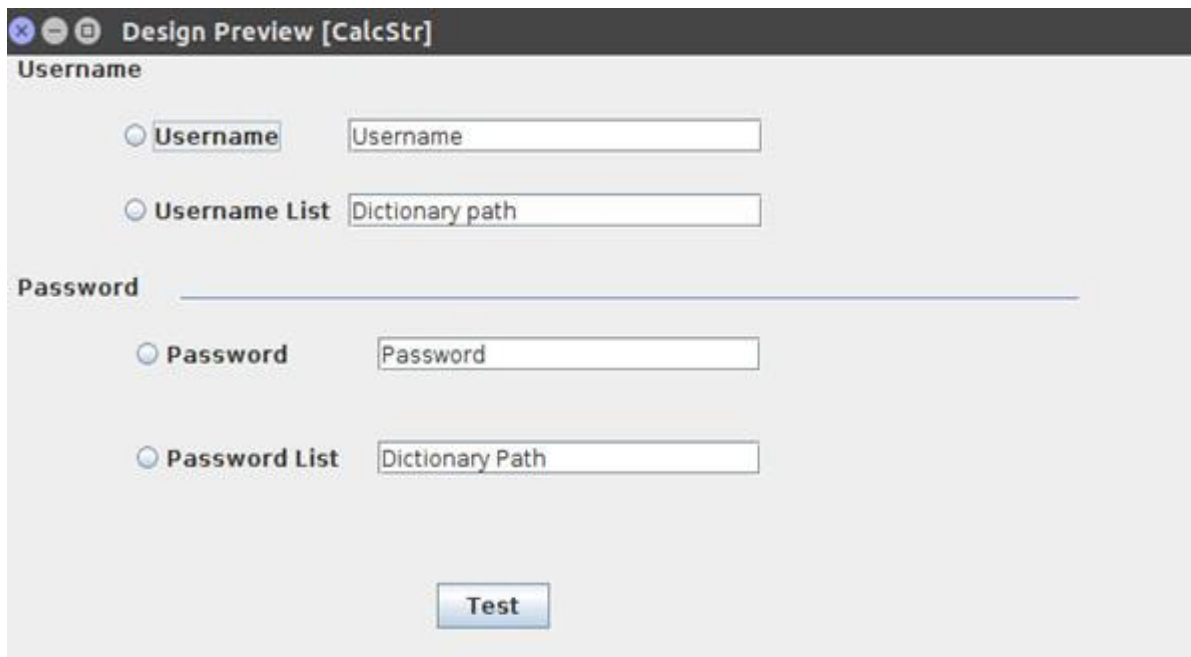
A screenshot of a dialog box with an orange border. It contains three labels on the left: "Mininum no. of characters", "Maximum no. of characters", and "Possible characters". Each label is followed by a text input field. The "Possible characters" field contains the text "iHIJKLMNOPQRSTUVWXYZ". Below these fields is an "OK" button.

Mininum no. of characters

Maximum no. of characters

Possible characters

OK



A screenshot of a window titled "Design Preview [CalcStr]". It has a dark header bar. Below the header, there are two sections: "Username" and "Password". Each section has two radio button options. The "Username" section has "Username" and "Username List". The "Password" section has "Password" and "Password List". Each radio button is followed by a text input field. The "Username List" field contains "Dictionary path". The "Password List" field contains "Dictionary Path". At the bottom center is a "Test" button.

Design Preview [CalcStr]

Username

☐ Username

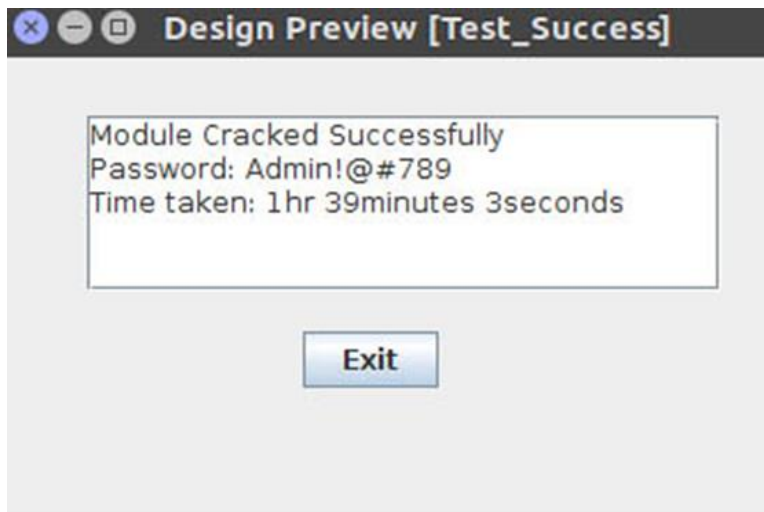
☐ Username List

Password

☐ Password

☐ Password List

Test



7. Legal Requirements

This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

The developers of the software are not responsible for any harm direct or indirect conducted using the software. The responsibility of morally and legally correct usage of the software lies solely on the user. The laws regarding the same vary from country to country, and all users are strongly recommended to refer the laws valid in their country to avoid any legal issues.

Appendix A: Glossary

Sand-boxing: Security mechanism for separating running programs, in execution of untested or untrusted programs and code, without harming the host machine or the host system.

Dictionary: A text file with a very large set of words/known passwords to generate potential passwords.

Brute-force attack: Trial and error method, which tests all the possible combination on something to get access to a particular resource.

Appendix B: Analysis Models

1. Usecase Diagram

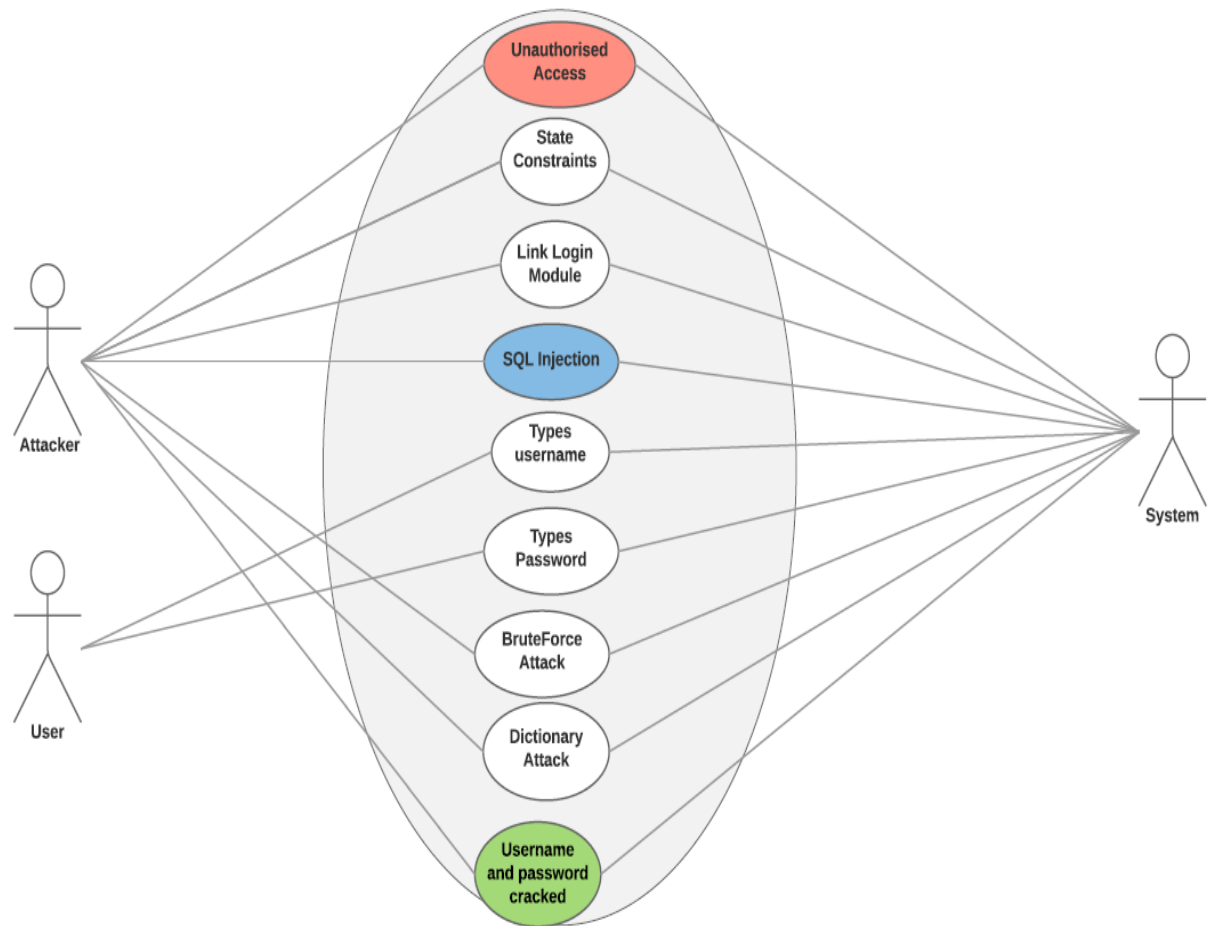


Fig.1 Usecase Diagram

2. Activity Diagram

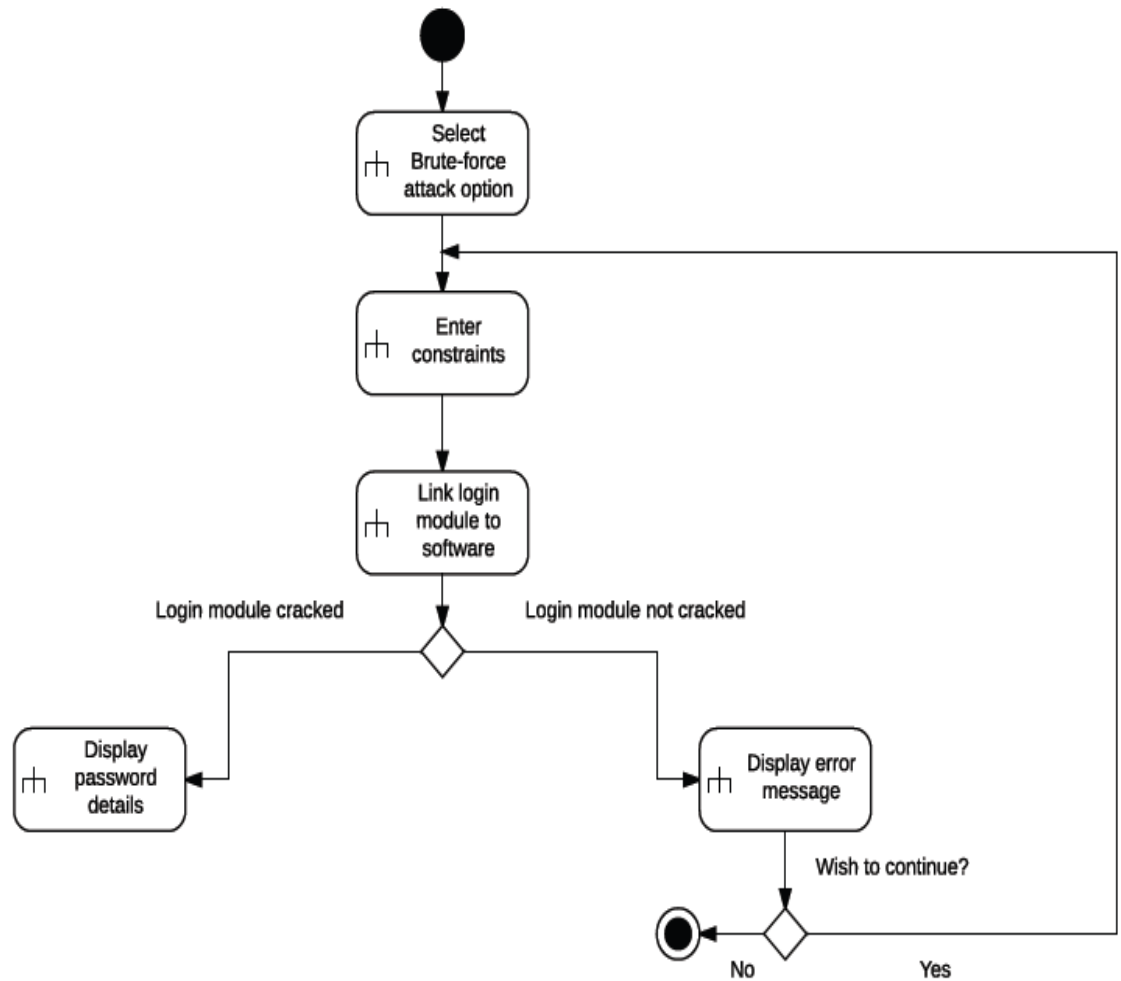


Fig.2 Activity diagram for Brute-Force attack

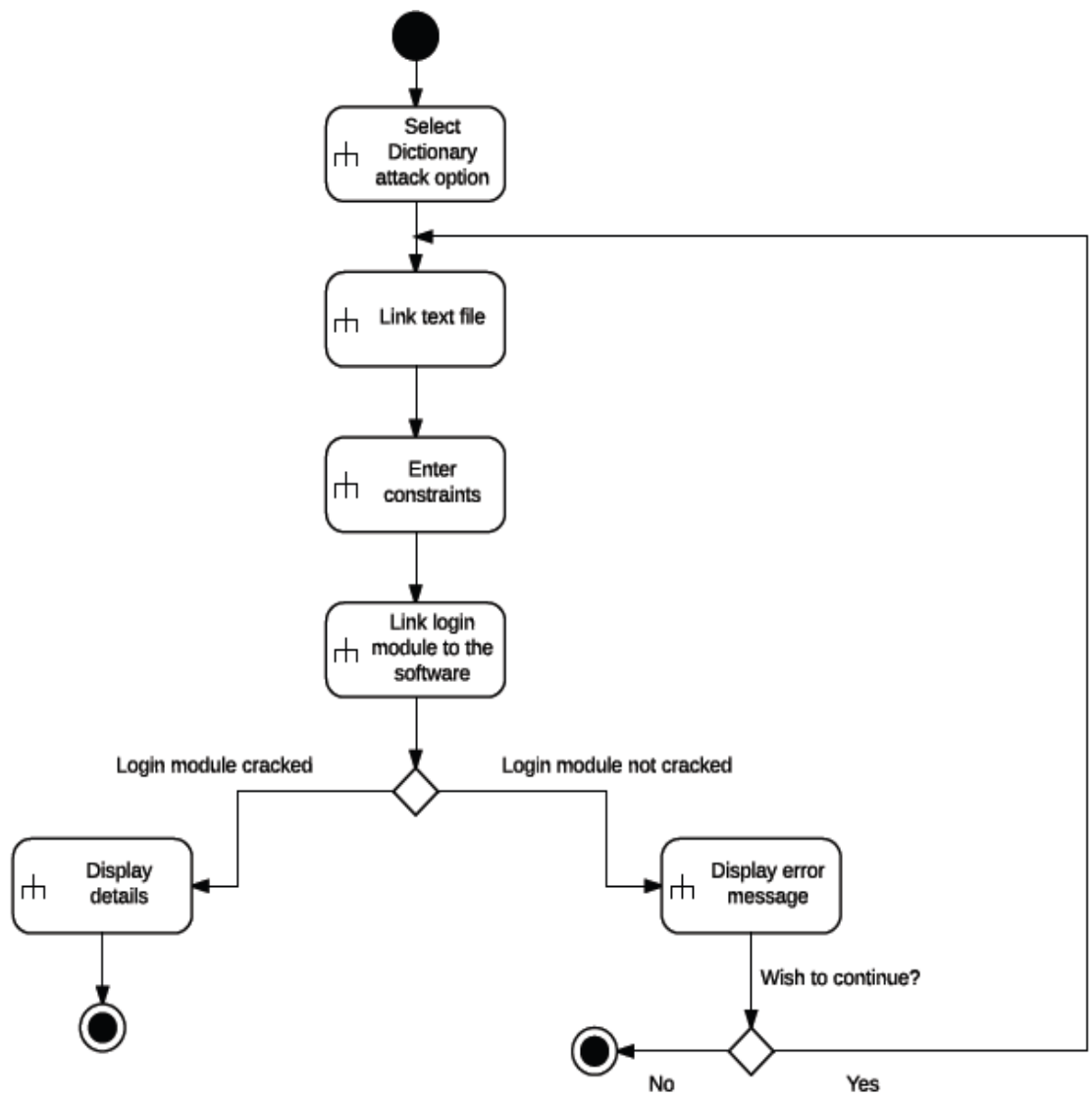


Fig.3 Activity Diagram for Dictionary attack

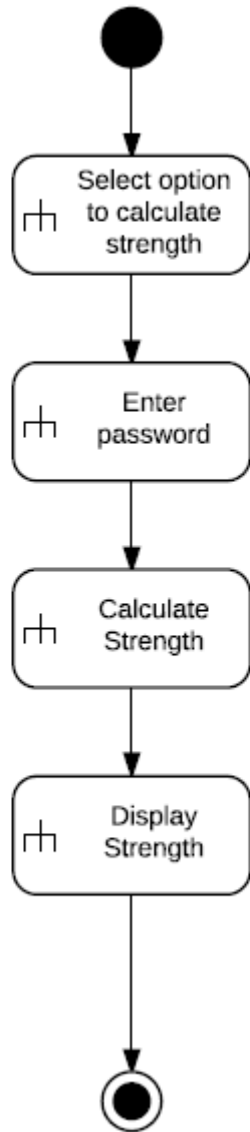


Fig.4 Activity diagram for strength calculator

3. Class Diagram

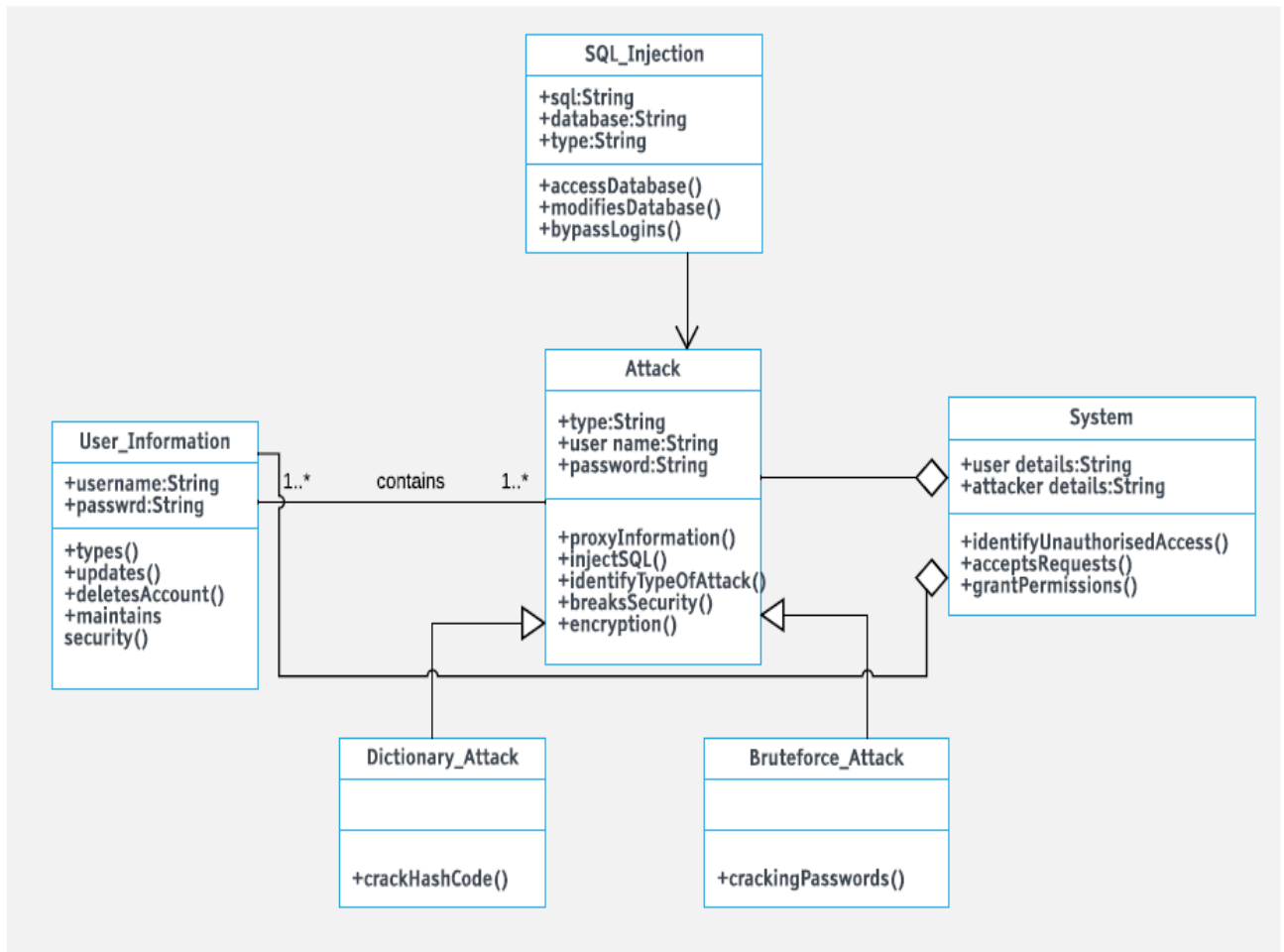


Fig.5 Class Diagram

4. Sequence Diagram

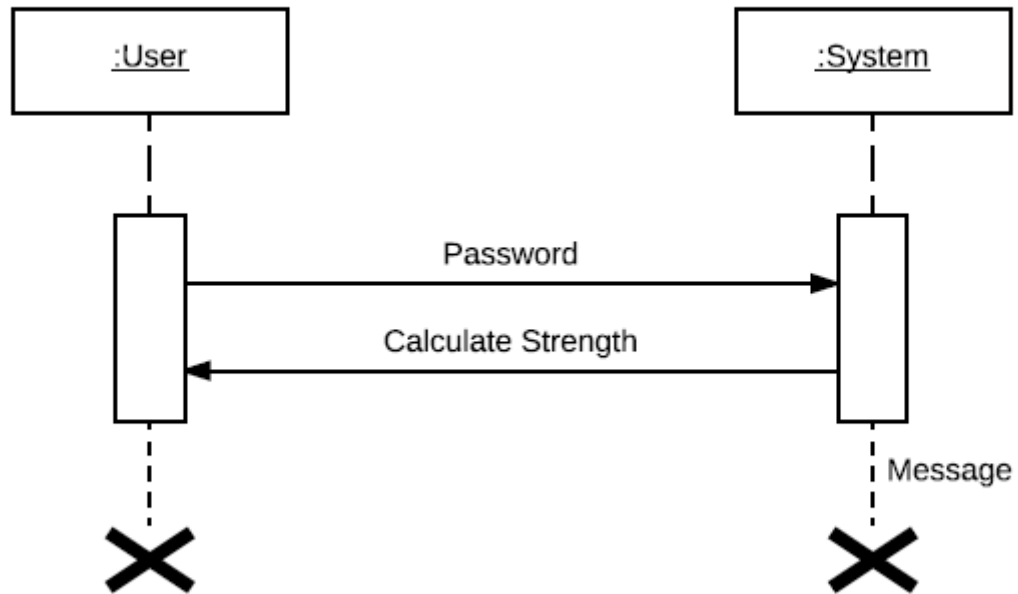


Fig.6 Sequence diagram for password strength checker

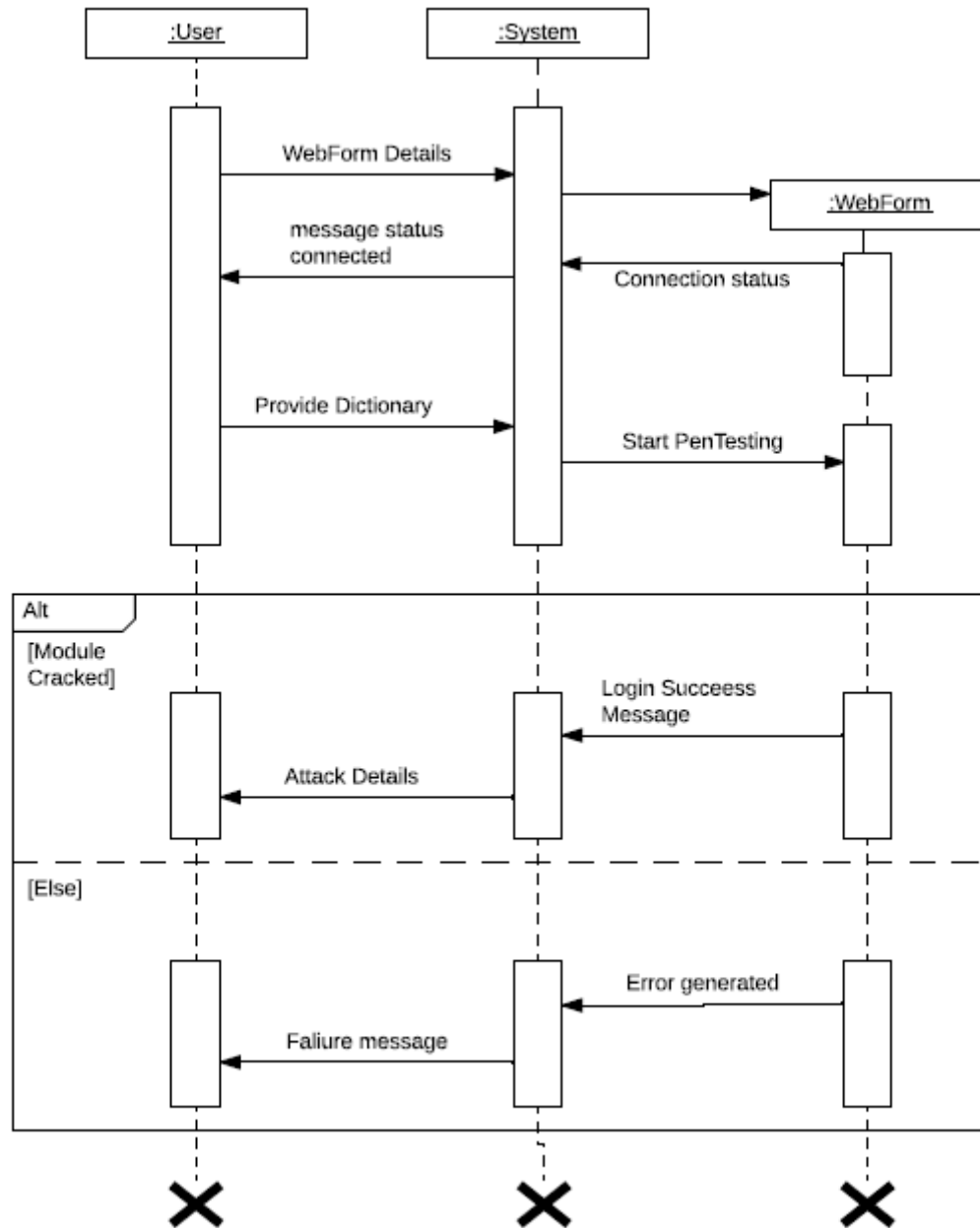


Fig.7 Sequence diagram for Dictionary attack

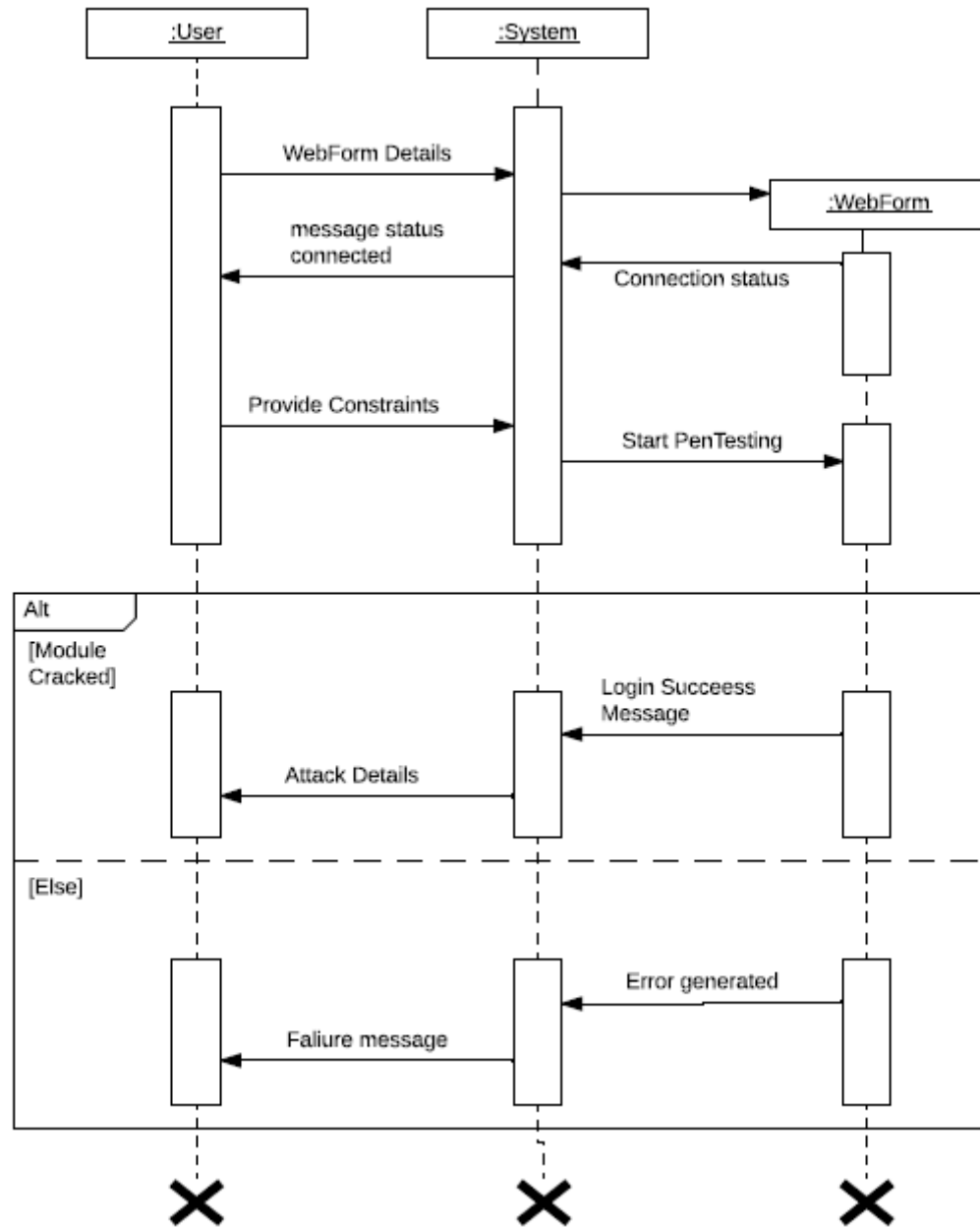


Fig.8 Sequence diagram for Brute-force attack

5. State Diagram

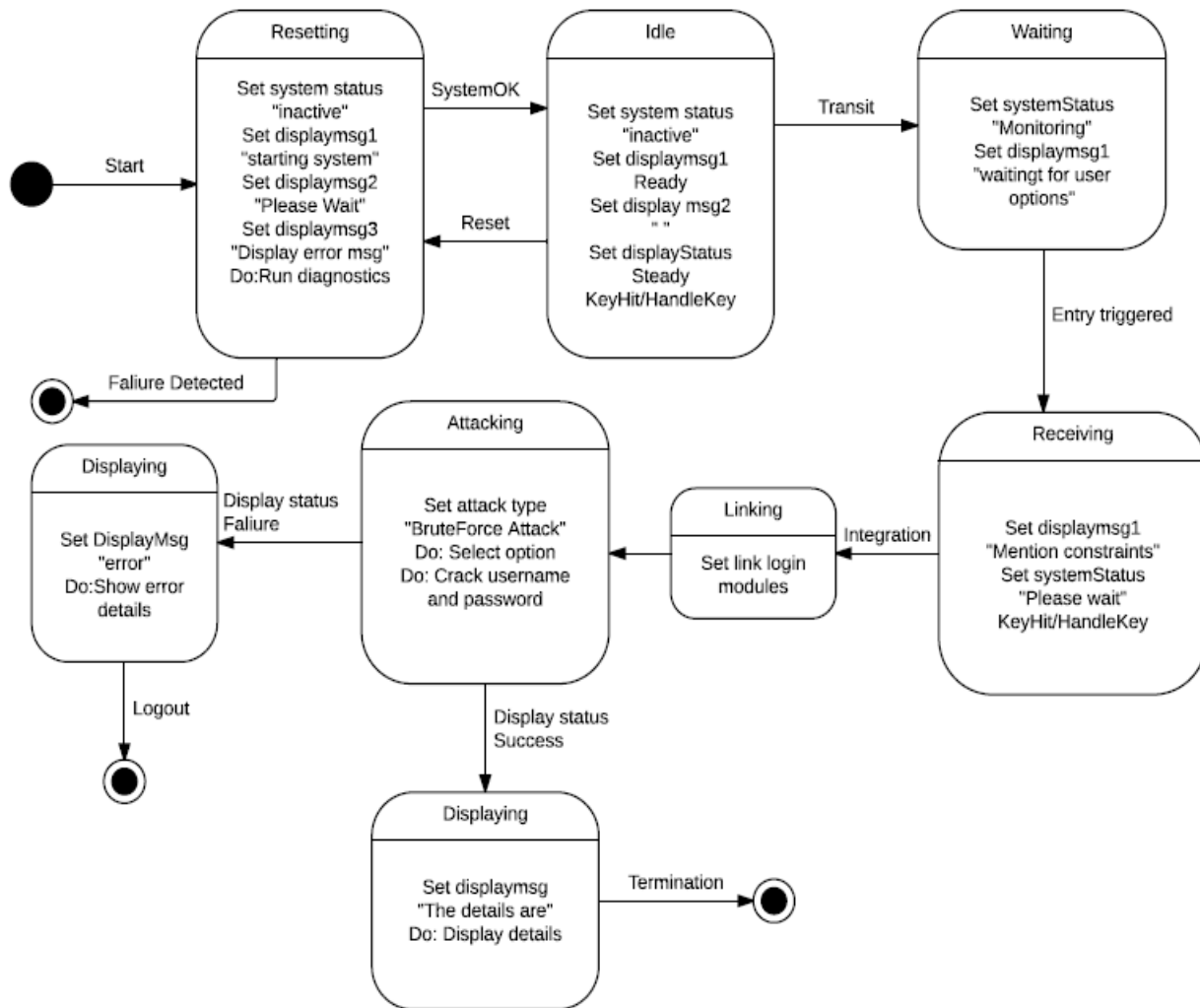


Fig.9 State diagram for Brute-force attack

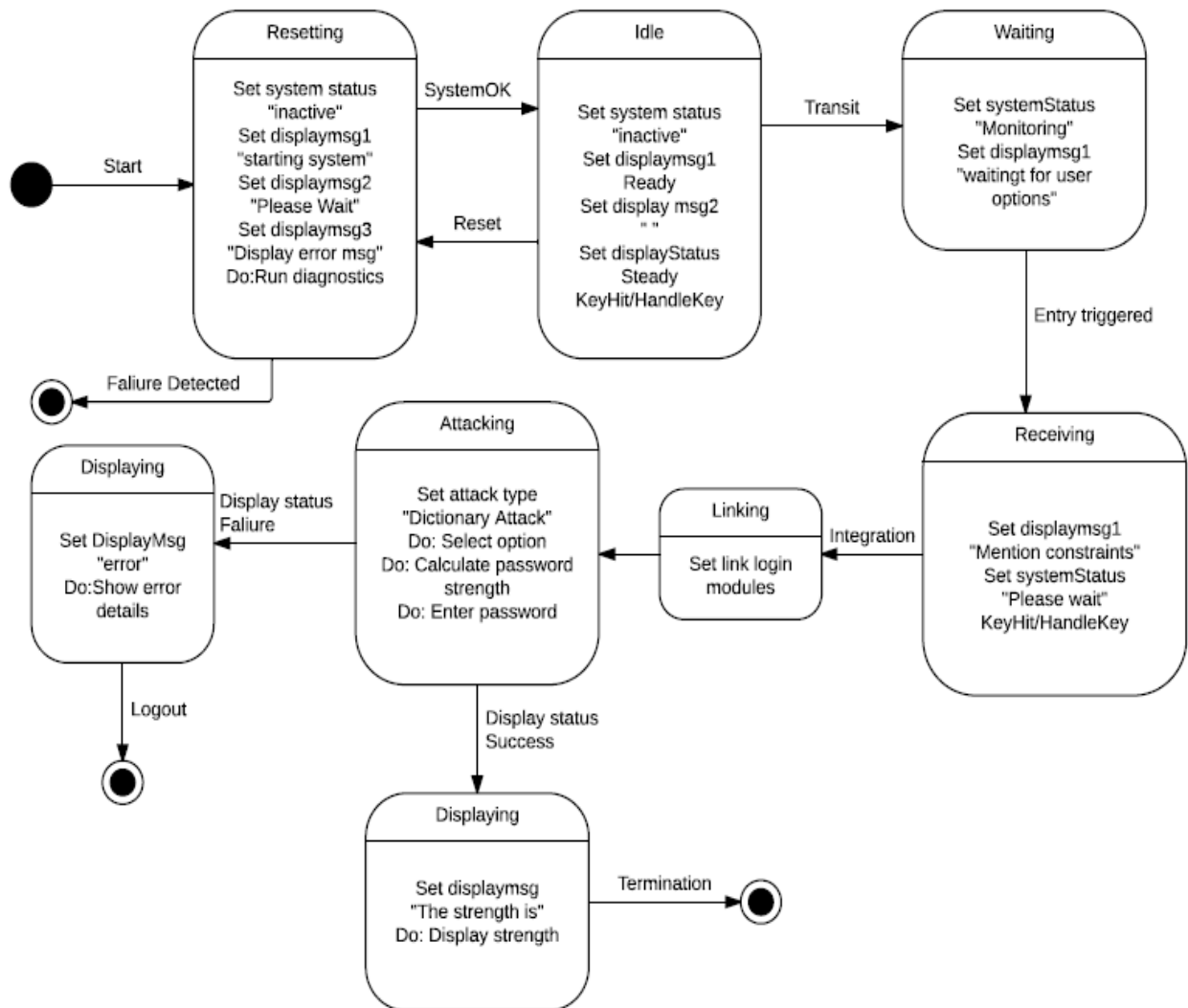


Fig.10 State diagram for dictionary attack

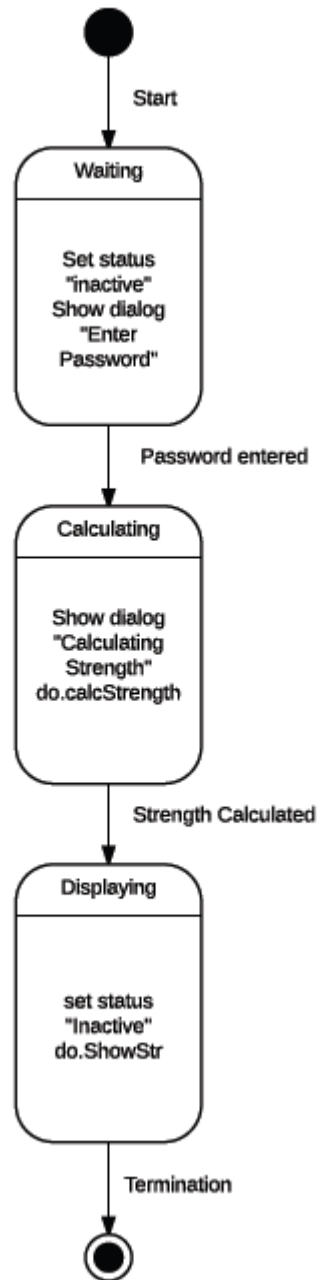


Fig.11 State diagram for strength calculation