# Social Engineering Prevention Training Tool - Project Proposal

Saleem Manjoo
Department of Computer Science
University of Cape Town
manjoosaleem@gmail.com

Michael Pepper
Department of Computer Science
University of Cape Town
mikejpepper@gmail.com

Marcel Teixeira
Department of Computer Science
University of Cape Town
marceltex@gmail.com

## 1. PROJECT DESCRIPTION

The field of information security is a fast growing discipline, with the protection of personal information being of vital importance [6]. Hackers are constantly seeking out new ways to exploit different aspects of computer systems [1], with one goal being the retrieval of sensitive personal information. To counter-act this, technological safeguards are developed, ideally mitigating the possibility and impact of such threats. This is a continuous cycle, leading to future attacks being more complicated and having to explore different avenues of attack. Furthermore organisations, governments and individuals are becoming increasingly aware of the threat of such technology-based attacks and are hence investing in better security technologies [1]. For this reason, some attackers (Social Engineers) have shifted their focus to exploit the new weakest link in the information security system - the user [6, 7]. This is achieved through the use of psychological ploys which compromise the user's emotional state, hence allowing an exploit to take place [2, 4, 6]. This psychological manipulation can be performed using various techniques through multiple channels and mediums, however the overall goal is the same. By exploiting psychological vulnerabilities within users, social engineers can elicit responses and perform information gathering that would not be possible had the user been in a more stable state of mind [7, 2]. This ultimately leads to the attacker achieving a predetermined objective, often unbeknownst to the victim.

### 1.1 Project Significance

The problem arises as often individuals do not perceive themselves as potential victims of such attacks and hence are not aware of the types of techniques used [5]. This ignorance can be attributed to their lack of knowledge of the potential gains an attacker can receive from the information they possess. Individuals may have the mindset that the information in their possession is not of any value to anyone, so why should they attempt to protect it [5]? Furthermore, some individuals feel they would be able to detect potential social engineering attacks however the social engineer is skilled at exploiting human vulnerabilities via psychological triggers in order to foil human judgement and attain information [7]. The situation is worsened by a severe lack of social engineering (SE) prevention tools, leaving potential victims with little-to-no way of protecting themselves against attacks. Large companies may have training for staff occasionally, however there is no available tool that can be used on a daily basis in real-time to determine the actions that they should take for any given scenario. This project aims to fill that void by creating applications that can be used by untrained personnel to identify whether they are falling victim to acts of social engineering or not.

The applications to be developed will use the SEADMv2 framework (Figure 1) [5] developed by Francois Mouton from the CSIR, which will ask users a series of questions, the outcome of which determines their progression through the model. The result of the questioning will leave the user in a predefined state in the model, and will indicate if they are in an SE attack and the actions they should take. The SEADMv2 framework determines this by assessing the authority level of the person requesting the information, the sensitivity of the information requested and the nature of the request, all of which aid in identifying the requester's motives.

### 1.2 Project Issues and Difficulties

As with any project, there are potential problems that need be accounted for and ideally mitigated. The first major difficulty that will need to be dealt with throughout the project revolves around logistics. The project is being performed in collaboration with the Council for Scientific and Industrial Research (CSIR) based in Pretoria, and hence effective communication may be a challenge as meeting in person is not possible. While this project is not a typical software driven project, the CSIR does act as a client whose requirements need be fulfilled. Effective communication will be pivotal in this regard. The second major concern revolves around the ethics of the project, as participants will be subjected to acts of social engineering. The appropriate ethical requirements and standards will need to be upheld throughout the project, as well as adequate post-attack debriefing. This will be dealt with further in section 5. A lack of experience amongst the team may also be deemed an issue, however the guidance provided by the CSIR and project supervisor Tommie Meyer should adequately mitigate this risk. Lastly a potential difficulty of the project is making the solution generic enough to be used in multiple scenarios. To achieve this, it will be tested on a range of

ACM ISBN 123-4567-24-567/08/06.
DOI: 10.1145/1235

attack templates, ensuring the underlying detection model's coverage is adequate for real-life scenarios.

## 2. PROBLEM STATEMENT

Currently there are limited resources available to aid in the prevention of social engineering attacks. Couple this with a general lack of knowledge about such attacks and the techniques used within them, and the probability for successful SE attacks is understandably high. The research questions aim at identifying the degree to which social engineering attack detection models can mitigate the risk of successful social engineering attacks, and are as follows:

**(1) Can a web-based implementation of the SEADMv2 framework reduce the probability of a subject falling victim to a social engineering attack?**

**(2) Can a mobile application (Android application) that implements the SEADMv2 framework reduce the probability of a subject falling victim to a social engineering attack?**

Successful reduction in the likelihood of individuals falling victim to social engineering attacks through the use of different mediums will aid in mitigating the risk of such attacks and the impact associated with them. It will also decrease the exploitation of unsuspecting individuals and ensure the privacy of their data. In addition, companies could make it mandatory for employees to use an implementation of a social engineering detection model, in order to minimise the threat of social engineering attacks.

The aim of our project is to develop three independent components that will help to reduce the likelihood of individuals falling victim to a social engineering attack. Two of the components comprise of a mobile and a web application. Since one of our aims is a universal tool that can be used in any environment, this is the most applicable solution. The web application will be more focused towards professionals in office environments where they have constant access to a computer. Attacks to these personnel are more likely to attempt to exploit their connection to their company, rather than the individuals themselves. The mobile application will be tailored towards use in daily life where attacks are more focused on exploiting the individual. By making a mobile app that has a responsive and easy to use interface, it should increase the overall use of the system as it can be integrated seamlessly into any scenario. This will aid in testing the effectiveness of the SEADMv2 framework.

The third component will be an independent back-end layer which will be developed to allow both the mobile and web applications to make use of, as well as any other possible platforms developed in the future. The purpose of the back-end layer is to provide access to databases storing the SEADMv2 as well as the user data that will be used for both the applications. The back-end will be designed to be as modular as possible so that changes to the SEADMv2 can be made easily, without the need for any of the interfaces to be changed. The development of the back-end layer will be treated as a software engineering project.

## 3. PROCEDURES AND METHODS

This section identifies the methods and procedures that will be implemented during the project life-cycle.

### 3.1 Development Features

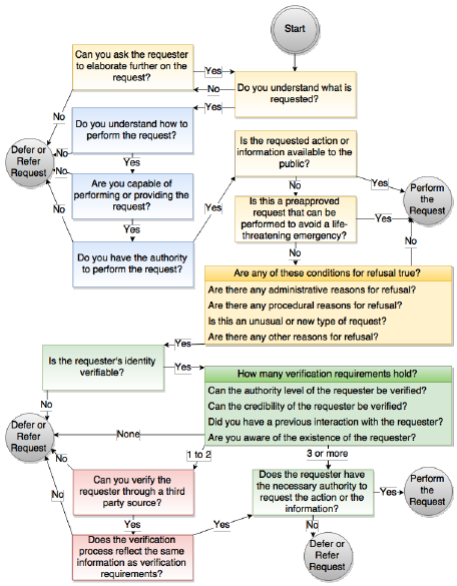


**Figure 1: SEADMv2 Mouton et al. [5]**

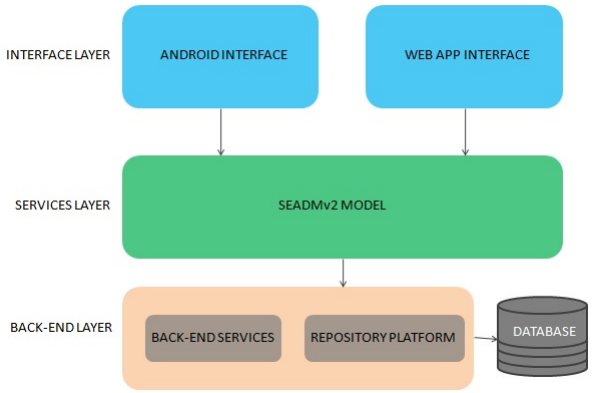


**Figure 2: Three-Layered Architecture Overview**

The effectiveness of the system revolves around the underlying model's ability to effectively detect, and hence prevent, social engineering attacks. This project will implement the SEADMv2 framework [5] as it is currently the most suitable for this application and is generic enough to ensure use within multiple scenarios. This model will mould the functionality of the system and will govern the prompts and questions the user is subjected to. Figure 1 depicts this model.

The system will be deployed through two separate mediums, namely as an Android app and a web-based app. The general structure can be observed in Figure 2.

*Interface Layer*: The interface layer is the medium through which users will interact with the system and navigate the SEADMv2 framework. The interfaces will be developed according to best practises relevant to their manner of interaction with the user, and taking usability into consideration.

*Services Layer*: The service layer connects the front end interface to the data-objects located in the back-end. The SEADMv2 will hence govern the information that is pre-

sented to the user and will determine the flow of events that the user is subjected to. Once the interaction is complete, this layer will co-ordinate the management of the information obtained.

*Repository Layer*: This layer deals with the storage and managements of information. This project will make use of an archival tool (discussed in the section below) as the basis of the development. This will ensure code integrity and adequate back-up procedures. The information attained from the user tests will be stored in a database which will also be managed in this layer.

## 3.2 Development Platform

The goal of the system is to have two independent interfaces that enable the same interaction with the SEADMv2 model, as well as an independent back-end layer from which both the mobile and web applications utilised. The mobile app will be developed using Java and will be able to work on any Android phone. The web-app will be developed using PHP and JavaScript (with Bootstrap), which will allow for it to work in an Web browser. The repository layer will be implemented using Git. The back-end layer will be developed using the Flask Framework and will be an independent component for which the current interfaces as well as any other interfaces developed in the future can utilise.

## 3.3 Implementation Strategies

The Rapid Application Development (RAD) methodology will be adopted for the SEPTT project as it is iterative and highly responsive to change. This will enable constant reviewable of progress and the direction the project is heading, whilst taking stakeholder concerns into consideration. The reactive nature of RAD is suitable for this project as should requirements change during the development process, existing work will not need to be re-done but rather altered to meet the new requirements.

Following the RAD methodology, the functionality of the system will be broken down into individual elements that can be prioritised and assigned to different members of the team. Features will be developed into prototypes, assessed at weekly meetings and then altered as necessary. A User-Centred Design approach will be adopted throughout the interface development, to ensure that the resulting system meets the requirements of the CSIR and is useful to prospective users. By involving users and stakeholders in the development of the interface, the resulting interface will be more natural to use as it conforms to user expectations, user satisfaction will be higher as user preferences can be incorporated, and design decisions can be made in consultation with the people who will actually use the system. These factors will all contribute to the uptake and continual usage of the apps.

## 3.4 Evaluation Methods and Acceptance Testing

Once developed, the overall success of the system will be assessed in multiple ways. Firstly user testing will take place whereby users will be subjected to certain scenarios and will have to use the system to progress through the SEADMv2 model. The scenarios will be provided by Francois Mouton and are composed of a series of questions that a requester will ask the user of the system. The user will then consult the system to determine how to respond to each request, and hence traverse the model. The result will inform them of the actions they should take and whether they are being subjected to social engineering or not. This will yield results that will either verify the ability of the SEADMv2 to detect and prevent social engineering attacks, or may indicate that the model is insufficient at detecting real-world attacks in this context.

The envisioned number of users that will be tested is between 30 and 50, and will be composed of university students. To add an incentive for participation, payment will be provided to participants. The funds for this payment will be raised by the project team in consultation with the CSIR, with the goal to pay participants roughly R30. Advertisements will be placed around UCT to recruit said participants, as well as word-of-mouth.

Secondly, software engineering metrics will be used to assess the success of the project. This will be coordinated through the use of software management tools (e.g. Trello) which will provide a universal record of the tasks required by each person and their completion status. The time taken to complete these tasks versus the estimated time will be assessed, along with whether the initial scope of the project was met or if functionality needed be reduced due to time/cost constraints.

To assess the success of the individual applications at preventing SE attacks, the user results will be analysed. To do this, users will first be placed in a scenario where an attacker attempts to exploit them personally. They will respond to the requests of the attacker without the use of the mobile application and the result will be recorded. They will then be placed in the same scenario and asked to respond to the attackers requests, however with the use of the mobile application. These results will also be recorded.

The user will then be placed in a different scenario where the attacker attempts to exploit their relationship with another party e.g. the company they work for. They will be asked to respond to the attackers request without the use of the web application at first, and then with the use of it. These results will be recorded. The results of users with and without the use of the two applications will be compared, and will indicate whether each application is effective at preventing users from falling victim to SE attacks or not. A comparison can also be made between the results from the two applications to determine which is more effective at preventing social engineering attacks.

For the back-end and the software engineering aspect of the project, 3 different methods will be used to evaluate whether or not it is successful. Firstly, the requirements that were outlined for the design of the back-end have to be met. These requirements include that the back-end be completely independent from the other layers, as well as being as modular as possible to easily accommodate any changes made to the attack detection model. Secondly, the software engineering metrics mentioned above will be used to assess the development of the back-end throughout its development. Thirdly, unit tests will be written and used to determined whether all code that is written works as intended.

## 3.5 Research Contribution

The resulting system will provide a free and easy to use tool for users in a multitude of contexts, which will help them prevent themselves from falling victim to social engineering attacks. The results will validate the ability of the

SEADMv2 to detect and prevent social engineering attacks in real-world contexts, as well as indicating which medium is used more effectively to prevent social engineering attacks (mobile or web-based).

# 4. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

As user testing on the applications developed will be conducted, ethical clearance from UCT will be have to be obtained. While there are ethical concerns regarding the use of subjects in social engineering research and since this research entails testing awareness of social engineering and the susceptibility of people towards it, all participants will be informed that they are in fact taking part in research regarding social engineering and thus the only special consideration during the user testing phase of the project would be to debrief the participants after the testing has taken place.

Everything developed in this project will be developed as open source and all code, deliverables and final applications will be made publicly available.

# 5. RELATED WORK

The Social Engineering Attack Detection Model version 2 (SEADMv2) by Mouton et al., 2015 [5], is not the only detection model that has been proposed to detect social engineering attacks. There are other models that have been proposed, some use similar techniques to the SEADMv2, while others use a completely different approach.

It should be noted, however, that the SEADMv2 is a result of two iterations of the original Social Engineering Attack Detection Model (SEADM) proposed by Bezuidenhout et al., 2010 [2]. The original SEADM had states at the beginning and end of the model that required the user to describe their emotional state and evaluate the level of discomfort they are experiencing. In the first iteration of the SEADM, it was argued that it is difficult for a user to evaluate their own emotional state. In addition, if the user has been having a bad day it is sometimes impossible for that user to determine the level of discomfort they are experiencing. Psychological measures were introduced in order to determine the users emotional state and the level of discomfort experienced. These psychological measures are proposed by Mouton et al., 2012 [7] and they involve a series of tests that the user is required to take, which will be used to determine the user's emotional state.

Using neural networks to detect social engineering, as proposed by Sandouka et al., 2009 [8], involves using a feedback neural network with 4 input layer nodes, 2 hidden layer nodes and 1 output layer node. The neural network has to be provided with sufficient training data, before it can be used. Once trained, the user will be required to provide the data requested by the input layer nodes. Using this data, the neural network will determine whether the user is a potential victim of a social engineering attack or not.

The Social Engineering Defense Architecture (SEDA), proposed by Hoeschele & Rogers, 2005 [3], uses a voice signature authentication system to detect social engineering attacks. It achieves this by maintaining a database of voice signatures linked to each employee's personal details. If a caller claims to be someone he/she is not, the SEDA should detect this, since the caller's voice won't match the signature stored in the database. The SEDA system is a good approach to detecting social engineering attacks, in that it is completely automated and requires little input from the user. The downfall, is that attackers can trick the SEDA system by using voice modulation.

Sawa et al., 2016 [9], proposes a system that uses natural language processing to detect social engineering attacks. This system will only be able to detect textual based social engineering attacks, such as phishing emails. The system extracts the topic from each sentence in the textual message. The topic of a sentence, in this context, is the pair consisting of the main verb and its direct object. This topic is then checked against a topic blacklist. If the topic is found to be in the blacklist, the user is alerted of a potential social engineering attack. Detecting social engineering using natural language processing is another example of an automated system used to detect social engineering.

# 6. ANTICIPATED OUTCOMES

In this section, the anticipated outcomes of the project will be discussed in terms of its software, its impact in the field of social engineering, as well as what factors are key to the project's success.

## 6.1 System

As this is partly a software engineering project, two systems, a web application and a mobile application, will be developed. The mobile application will be a mirror of the web application and will have the same functions and purposes of the web application, albeit on a different platform and with a different interface.

These applications will act as training tools to combat social engineering by educating the user about social engineering and how to prevent it. Both applications will share an extensive database of usage data that will be captured and later used for statistical purposes. The applications will be heavily GUI reliant and be designed to be as easy to use as possible by the user.

## 6.2 Expected Project Impact

We aim to increase the awareness of social engineering as well as the potential impact of its threats. We hope to produce applications that are effective tools for combating social engineering, as well as provide them with ways to identify and prevent a social engineering attack.

## 6.3 Key Success Factors

The success of the project will be based on whether or not users of the applications are more capable of identifying and preventing social engineering threats after using the applications. In order to achieve this, both applications will require a large database of social engineering attack examples as well as interfaces that are designed to be as easy to use as possible.

# 7. PROJECT PLAN

## 7.1 Risks and Risk Management Strategies

The key risks as well as their consequences, mitigation strategies, monitoring plans and management plans are tabulated in Table 1.

## 7.2 Timeline

A Gantt chart illustrating the proposed timeline for this project is shown in Figure 3 and Figure 4. Due to the length of this Gantt chart it has been split across two pages. The diamonds on the Gantt chart illustrate project milestones.

## 7.3 Resources Required

In order to develop the software we will require computers that are capable of compiling the source code for the desired platform. Since the web application and back-system are rather generic, we should be able to use any computers for development. Due to the open source nature of Android, it is also possible to install the SDK and compile the source code on any operating system. The web site for the web application will be hosted by Afrihost and the domain is owned by the CSIR, therefore we will not require our own servers to host it.

The framework of the SEADMv2 will also be required in order to implement it. This will be provided to us by our supervisor, F. Mouton, who has documented all of this already. In addition, we will require users to test our applications. These users will be anyone from the general public, since anyone is at risk of falling victim to a social engineering attack. We will also require sample attack scenarios that we can provide the users with while they are testing the applications.

## 7.4 Deliverables

The deliverables for this project are listed below in the typical order that they will be completed.

- A literature review of the available literature on the topic of Social Engineering
- A project proposal document (this document)
- A project proposal presentation
- An initial feasibility demonstration
- An Android App that implements the SEADMv2 to help users detect whether they are the victim of a social engineering attack.
- A web app that implements the SEADMv2 to help users detect whether they are the victim of a social engineering attack.
- A back-end system that will be used to determine the effectiveness of both the Android app and the web app as well as compare the two apps.
- Unit tests, to test the Android app before it is released.
- Unit tests, to test the web app before it is released.
- Unit tests, to test the back-end system to ensure it works correctly.
- A summary of results obtained after tests were performed on users.
- A final project paper discussing the findings of our investigation.
- The source code of all apps developed.
- A demonstration of our project.
- A poster summarising our project.
- A web page describing our project.
- A reflection paper.

## 7.5 Milestones

The milestones and tasks required to achieve these milestones are outlined in the Gantt Chart in Figure 3 and Figure 4. The milestones are indicated with diamonds on the Gantt chart. Below is a list summarising the milestones for this project.

**Milestone A:** Literature review of previous work done on project topic. (26 April 2016)

**Milestone B:** Project proposal written document. (17 May 2016)

**Milestone C:** Project proposal presentation. (24 May 2016)

**Milestone D:** Initial software feasibility demonstration. (18 July 2016 - 22 July 2016)

**Milestone E:** Weighting for project marking decided. (17 October 2016)

**Milestone F:** Final complete draft of project paper. (18 October 2016)

**Milestone G:** Project paper. (28 October 2016)

**Milestone H:** Project code. (31 October 2016)

**Milestone I:** Project demonstration. (31 October 2016 - 4 November 2016)

**Milestone J:** Project poster. (7 November 2016)

**Milestone K:** Project web page. (11 November 2016)

**Milestone L:** Project reflection paper. (14 November 2016)

## 7.6 Work Allocation

Since this project consists of three core components, it was easy to allocate the work among ourselves. We allocated the work according to our strengths, so that development can be as efficient as possible. It was decided that Michael will develop the web app part of the assignment. He will integrate the web app into the Social Engineer South Africa web site (http://www.social-engineer.co.za). Marcel will develop the Android app. The Android app will be similar to the web app in that it will also implement the SEADMv2 in an attempt to help users detect social engineering attacks. Saleem will develop the back-end system of this project. This system will be used to store the database used by both the web app as well as the Android app. In addition, it will store results obtained from testing users. It will be designed to be as modular as possible to allow for changes to be made to the attack detection model, being used, without any changes being required to be made to any of the interfaces.

## 8. REFERENCES


[1] Abraham, S., and Chengalur-Smith, I. An overview of social engineering malware: Trends, tactics, and implications. *Technology in Society 32*, 3 (2010), 183–196.

[2] Bezuidenhout, M., Mouton, F., and Venter, H. S. Social engineering attack detection model: Seadm. In *Information Security for South Africa (ISSA), 2010* (2010), IEEE, pp. 1–8.

[3] Hoeschele, M., and Rogers, M. Detecting social engineering. In *Advances in Digital Forensics*. Springer, 2005, pp. 67–77.

[4] Mouton, F., Leenen, L., Malan, M. M., and Venter, H. Towards an ontological model defining the social engineering domain. In *ICT and Society*. Springer, 2014, pp. 266–279.

[5] Mouton, F., Leenen, L., and Venter, H. Social engineering attack detection model: Seadmv2. In *2015 International Conference on Cyberworlds (CW)* (2015), IEEE, pp. 216–223.

[6] Mouton, F., Malan, M. M., Leenen, L., and Venter, H. S. Social engineering attack framework. In *Information Security for South Africa (ISSA), 2014* (2014), IEEE, pp. 1–9.

[7] Mouton, F., Malan, M. M., and Venter, H. S. Development of cognitive functioning psychological measures for the seadm. In *HAISA* (2012), pp. 40–51.

[8] Sandouka, H., Cullen, A., and Mann, I. Social engineering detection using neural networks. In *CyberWorlds, 2009. CW'09. International Conference on* (2009), IEEE, pp. 273–278.

[9] Sawa, Y., Bhakta, R., Harris, I. G., and Hadnagy, C. Detection of social engineering attacks through natural language processing of conversations. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)* (2016), IEEE, pp. 262–265.

| Risk Condition | Consequence | Mitigation | Monitoring | Management |
|---|---|---|---|---|
| Absence of a team member. | Increased workload for remaining team members and possibility of missing deadlines. | Perform frequent health checks within the team. At the start of the project, ensure team members are committed to their honours degree and the project. | After each milestone, re-establish project commitment and long-term health. | Ensure that team member skills are shared to some degree. Ensure consistent coding practises are adhered to and code is commented thoroughly, this will make it easier for other team members to continue the work, should a team member leave. |
| Theft or loss of source code. | Needing to replace and re-code the software development that was lost. | Use a GitHub repository for all source code and push to the repository regularly. | Ensure each team member has been pushing their work to GitHub, at least once a day, if not, confront the team member who isn't. | If loss of source code or theft does occur, continue development from the latest version of the source code on GitHub. |
| Failing to meet milestone deadlines. | Losing reputation with the project supervisor and a loss of marks per day over the deadline. | Use realistic dates when creating the project schedule. Divide the project into smaller, manageable tasks, each with their own deadlines. | After completing each task, compare the schedule to the progress of the actual project. If major discrepancies exist, consider adjusting the scope of the project to ensure it is completed in time. | Alert the project supervisor well in advance, if the project is falling behind schedule and discuss possible scope changes that could be made. |
| Feature inflation - Too many features are added during the project. | Added features are out of scope and not required for the purpose of the project. | If any features are added during development ensure the project supervisor approves and these features are definitely required. Plan the project thoroughly before development to encompass all required features in the original plan. | During development constantly refer back to the project proposal document, to ensure that only the planned project is being developed and nothing more. | Communicate all features with the project supervisor and negotiate the removal of unnecessary features. |
| Team members dishonest about their skills or level of competence. | Incomplete features or missing features in the final version of the software. Project development time wasted on team members learning the required skills. | Full disclosure of personal and technical skills discussed amongst the team members at the start of the project. | As development of each feature begins, re-evaluate the skills needed for completion of that feature and ensure that one of the team members has that skill or plans to obtain it. | If the risk is detected early on, ensure the skills that are needed, are learnt early on, leaving enough time to use the skill to implement the required feature. |

**Table 1: Risks and Risk Management Strategies**

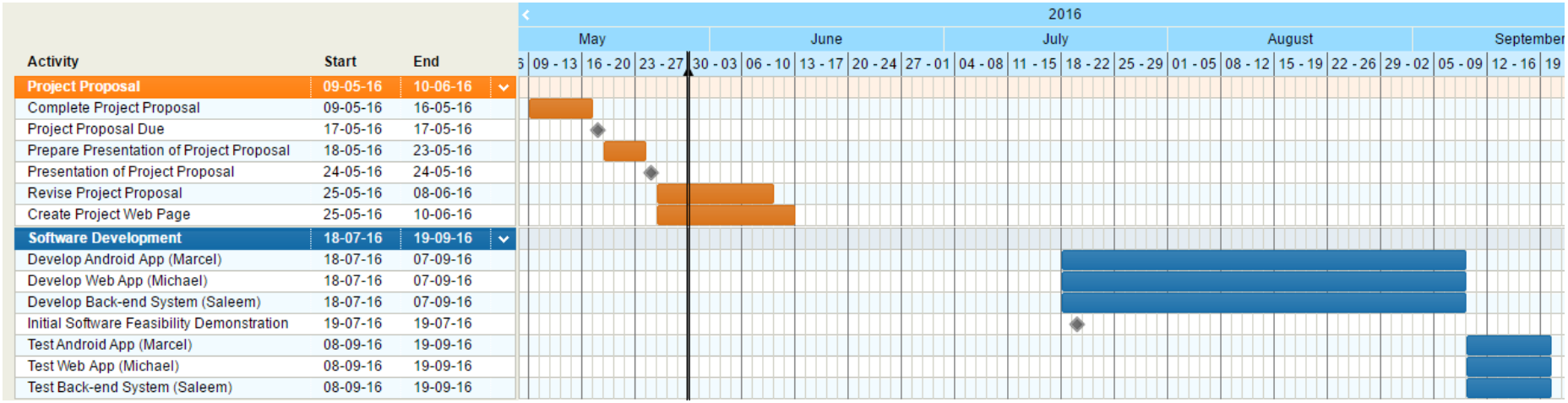


**Figure 3: Gantt Chart showing timeline of the first half of the project**

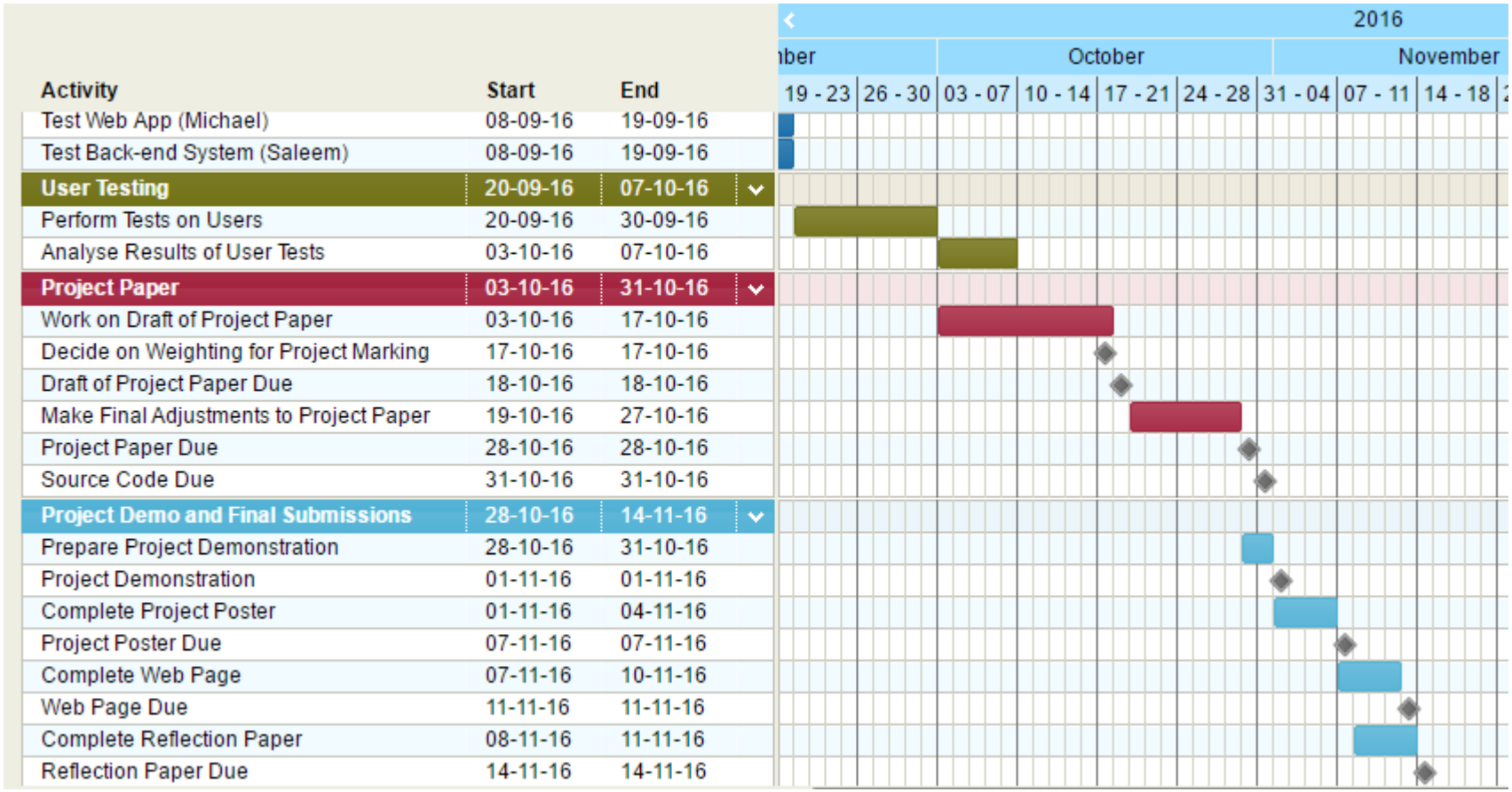


**Figure 4: Gantt Chart showing timeline of the second half of the project**