Software Testing Plan
Version 1.0

01 April 2022

SuperGeeks

CS 486C
Capstone Experience

Team Members: Tim Giroux, Kyle Austria, Gustavo Valencia, Pengfei Liu

Clients: Dr. Marco Gerosa, Dr. Igor Steinmacher

Mentor: Felicity Escarzaga

Table of Contents

1. Introduction

Software engineering and development has constantly evolved over the years. It is becoming more complex and advanced. With this, research is essential to improve development processes, architecture, and the tools used for development. Quality research will help the software industry by providing valuable information that can drive the success of projects around the world. It will provide researchers and developers alike with highly valuable insights that can be used to improve software that many modern industries use today.

GeekSurvey will serve as a web application that will facilitate software engineering research coordination between researchers and participants. Unlike other recruitment platforms, GeekSurvey will be able to filter out participants based on their qualifications and their specialized profiles. This will provide the necessary information for selective recruitment and enhance the clients' business process.

Software testing is needed to ensure that GeekSurvey does what it is intended to do. Simply, software testing is the process of evaluating and verifying an application and its functionalities. The goal is to make sure that the application works as intended and that there are no bugs that went unnoticed. By testing software, performance can be improved and the development cost can be reduced since errors will be detected earlier on in a project. Overall, software testing will improve quality and understanding of the software, reduce costs, and help produce software that functions cleanly without problems.

GeekSurvey will be tested in a variety of ways. Unit testing will be conducted to detect any errors that may occur within each part of the codebase and determine the cause of those errors. Regarding GeekSurvey, the current codebase contains some elements that can be tested using unit testing, but should ideally be refactored to better accommodate unit testing in the future. Then, integration testing will be performed to determine how external platforms, such as Google Forms and the Paypal API, interact and exchange data with GeekSurvey. The goal is to see if any difficulties or problems arise when using the external platforms in relation to GeekSurvey. Finally, usability testing will be conducted to see how end users use GeekSurvey and its features. There are features of GeekSurvey, from account creation to study enrollment, that will need to be tested multiple times with various accounts to completely determine if there are any functionality issues.

The testing plan for GeekSurvey is designed for a web application that will facilitate numerous end users interacting with each other, as well as external platforms. The size of GeekSurvey's codebase is quite large and dividing them into parts will make it easier to detect errors and add new code in an organized way. The tests that will be conducted on GeekSurvey will aid in producing a platform that will not only do what it is intended to do, but also enhance the scalability for future development and updates.

2. Unit Testing

Unit testing is a type of software validation that involves breaking up a larger program into its constituent parts, and testing each part individually. This is used by software engineers to catch any problem that might be caused by new changes to a codebase. When a codebase has significant unit testing coverage, engineers have a great tool for determining the cause of an error. They can see exactly what unit tests are failing, and then go investigate that specific part of the codebase.

It is difficult to accomplish significant unit test coverage in a code base that is not designed to be tested. In the long term, GeekSurvey would need significant refactoring to accomplish significant testing coverage. That being said, there are still many unit tests that can be implemented for GeekSurvey as is.

2.1 Testing Before Refactoring

GeekSurvey has a number of utility functions that are used in the views and models logic. These include email utilities, payment utilities, and an enrollment utility. The enrollment utility, **can_enroll**, can be tested rather simply with some sample study and profile objects. Tests can use some sample objects for which can_enroll should return True, and some for which it should return False. GeekSurvey can have a test for each enrollment criteria to ensure good test coverage. The other utilities would be slightly more difficult to test, because they involve networked inputs and outputs.

For example, there is an email utility that is used to send email notifications to users depending on their eligibility, and whether they have opted into email notifications. This function is difficult to test because its correctness relies on its ability to cause effects over Google Mail SMTP, which is external to the GeekSurvey system. The same predicament applies to some payment utilities. Instead of unit testing, these items must be treated as integration testing.

GeekSurvey can test for boundary values in our database objects that have minimums and maximums, such as the various age fields. The GeekSurvey codebase assumes that no user is older than 150 years old, and no younger than 0. It would be worthwhile to write unit tests to validate a proper response for these boundary values, as well as for negative values and values above 150.

GeekSurvey can also test for erroneous inputs, such as HTTP requests to access or edit application resources that should be only accessible to certain users. An example of this is if user X tries to edit user Y’s profile. In this case, the request should essentially be denied, and no database information should be altered. The same applies to studies, which can only be edited or managed by their owners. Another example of this is in claiming account funds. Of course, it would be highly problematic if user X claims the funds from user Y’s profile. All of these areas of restricted access can be validated by unit testing.

2.2 Refactoring GeekSurvey

For unit testing to be properly implemented in a codebase, the codebase must be modular and loosely coupled. It is often challenging to implement unit tests for a codebase that was developed without unit tests in mind. This is one reason why many software projects decide to use *test driven development*, which requires tests to be implemented for every new feature before the feature itself is implemented.

As of March 2022, GeekSurvey has not been developed using test driven development. Unit testing for GeekSurvey would be best implemented starting with a large-scale refactor of the codebase. Much of the backend logic could be preserved and transformed into a tightly specified REST API, and the frontend could be rewritten as a React application or something similar. After this kind of refactor, sample API input could be tested with clearly defined truth output. Such a refactor would not only improve the potential for testing coverage of the codebase, but would also improve the codebase generally.

3. Integration Testing

Integration testing is the process of testing the interaction between major modules and components. It focuses on the correctness of the interactions and data exchanges between modules. Oftentimes, modules need to interact with third party tools or APIs to test whether the responses generated are as expected. The purpose of the integration testing is to expose faults in the interactions between integrated units. Through integration testing, developers can make sure that the modules/components work fluidly as one system.

GeekSurvey is a web application that will facilitate software research coordination between researchers and participants. It helps researchers conduct more professional research and obtain more high-quality data. From this perspective, a safe and stable external survey platform is needed. After doing feasibility studies, it was found that Google Forms can meet GeekSurvey's need for an external survey platform very well. Another important component of GeekSurvey is the external payment platform, which will allow GeekSurvey researchers and participants to manage their account funds when creating or completing studies. The Paypal Sandbox API is a mature payment utility that will satisfy GeekSurvey's need for a payment platform. In order to make sure that Google Forms and the Paypal Sandbox API have correctly interacted with GeekSurvey, these items must be tested using integration testing.

3.1 Google Forms

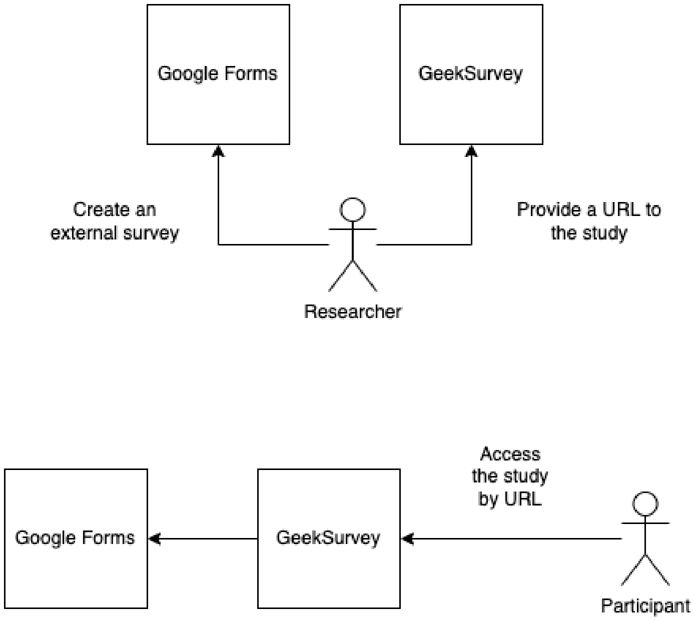

Figure 3.1

Google Forms is a third party survey platform that interacts with GeekSurvey to make it possible for researchers and participants to create and participate in studies. The survey forms are crucial to conducting the research, as it is through these forms that researchers can access the required data. In order to test whether Google Forms correctly interacts with GeekSurvey, we will need to demonstrate that a researcher can create a Google Form that contains a completion code to be used on GeekSurvey. Then, within the GeekSurvey study object, the URL to the previously mentioned Google Form will be entered in the respective field. As shown in Figure 3.1, a participant on GeekSurvey would test for successful integration by completing the linked Google Form in the GeekSurvey study object and entering the completion code from the Google Form into the GeekSurvey study object.

3.2 Paypal Sandbox API

The Paypal Sandbox API is an efficient and concise payment method interface that allows for integration with other platforms using mock data and transactions. By using the Paypal Sandbox API, GeekSurvey can implement an external payment system. Since GeekSurvey is a web application that aims to solve the researcher-participant coordination problem in software engineering research, it is important for GeekSurvey to allow researchers to incentivize qualified participants to participate in their studies.

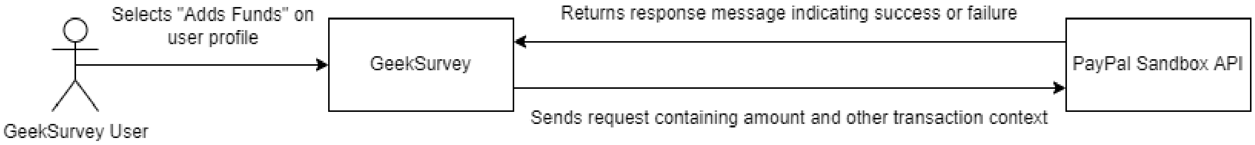
Selects "Adds Funds" on user profile
GeekSurvey User
GeekSurvey
Returns response message indicating success or failure
Sends request containing amount and other transaction context
PayPal Sandbox API

Figure 3.2

To test whether the Paypal Sandbox API correctly interacts well with GeekSurvey, funds will need to be added to the user's account. The user would navigate to the fund profile page, and click on the “Buy Now” button. The user will then be redirected to the PayPal Sandbox payment portal, where they will enter mock credentials and follow the payment process. Upon confirming the purchase, the user should be redirected back to the merchant (in this case, GeekSurvey) and receive either a success or failure message. The user should then note changes in their account balance. The PayPal Sandbox API can be tested again by having the users move funds out of their GeekSurvey accounts and to their mock PayPal accounts. The user would do this by clicking the “Remove Funds” button and by entering the email of the PayPal account they would like to transfer the funds to. As shown in Figure 3.2, one way a user on GeekSurvey would test for successful integration is by navigating to the user profile page, selecting the “Add Funds” button, following the PayPal Sandbox API workflow, and then checking their account balance for the purchased funds.

4. Usability Testing

Before describing what usability tests are, why they are needed, and how they work, it is important to introduce some helpful context surrounding usability tests. Unlike the other types of tests discussed thus far, usability tests are not their own standalone category of tests, but rather a subcategory that falls under the umbrella of acceptance tests. Acceptance tests typically focus on testing whether a system meets some acceptance criteria. These types of tests are generally done from a graphical user interface (GUI) and are considered a form of black box testing, as a system's functionality is tested by observing its external behavior.

Now that we have introduced how usability tests are a type of acceptance testing, we can dive deeper into what exactly usability testing is, why it is needed, and how it works. Usability testing aims to test the general usability of a software product from the point of view of an end user. It is through usability testing that the interaction between end users and the many features of a software product are observed. For example, how a user interacts with a software product's account creation process is something that would be observed and noted during usability testing.

Usability testing is needed as it helps give developers of a software product valuable insight into how end users currently perceive the product. This is especially important as it can be difficult for developers to see the product they are building from the point of view of an average end user, as they are experts that understand the product inside and out. Usability testing can also help identify changes required to improve the usability and overall satisfaction of a software product.

Finally, usability testing can work in a variety of ways. While it can be done in a formal laboratory setting, there is typically no such need for a formal setting. Oftentimes, usability testing is simply conducted by having someone observe the end user and take notes. Additionally, there is also no explicit need for this testing to be synchronous or in-person, as it is common for usability testing to be conducted remotely. The usability testing also does not necessarily need to be moderated. The way the usability testing is conducted should be tailored in a way that collects data in the most useful way, and is dependent on the software product being tested.

4.1 Usability Testing Context

As mentioned previously, a usability testing plan should be tailored to the individual software product being tested. As such, it is important that all relevant features of the product that could impact the design of the usability testing plan are laid out. Some of these features may include the types of users using the software product, the software product's novelty, and the background of its end users. Once these features are identified, they can be used as the context that shapes the usability testing plan.

To begin, the types of users that would use GeekSurvey are those looking to conduct and participate in software engineering research studies. These users have likely used other competitors, such as Prolific.co, and found that they did not satisfy their needs. This may be due to lack of features tailored for software engineering research. This is where the novelty of GeekSurvey comes in. GeekSurvey aims to provide many of the same features that its competitors offer, while also tailoring features to better conduct software engineering research. This can be seen through the profile fields for GeekSurvey accounts, which are used for study discovery filtering. Finally, the background of GeekSurvey's user base is varied but technical. This can be seen through the occupation profile field, which allows users to select an occupation that best fits their occupation. While all the occupation options are varied in terms of what each role does, they all include a technical aspect, as they can be considered subfields of software engineering and computer science.

Regarding how these features will shape the usability testing plan for GeekSurvey, the types of users (researchers and participants of software engineering research) will shape the plan by having observed users utilize both the research and participating aspects of GeekSurvey. The usability testing plan will be shaped in a way that allows users to give feedback on what they like about the current research and participation features, and what they would like to see from both features. The novelty of GeekSurvey will shape the usability testing plan by having observed users provide feedback on GeekSurvey's software engineering research specific features, especially in comparison to competitors. The background of end users will shape the usability testing plan by requiring that those who take part in usability testing have a technical background that is relevant to GeekSurvey.

4.2 Plan for Usability Testing

Now that context has been established for how features of GeekSurvey will impact the design of the usability testing plan, we can now dive deeper into what the plan will include. The types of tests that those involved in usability testing will be subjected to will be basic tasks on the platform. The tasks outlined in the usability testing plan would be as open-ended as possible, as it would be beneficial to see what end users intuitively do when prompted. For example, some tasks tests users could be prompted to do include:

- *Sign up/Sign in:*
  Test users will sign up with an email address or sign in with a social account like GitHub or Google. The test user's method and overall experience with this task will be recorded.

- *Edit User Profile:*
  Test users will edit their profile upon account creation and add any information they find relevant. The test user's overall experience with this task will be recorded.

- *Create a Study:*
  Test users will create a study object on the platform. This task may also direct users to create a mock survey on Google Forms for use in the study object. The test user's overall experience with this task will be recorded.

- *Fund a Study:*
  Test users will first fund their accounts. They will then return to their previously created study and add funds to it. The test user's experience with the PayPal payment system and with moving funds from their profile to the study object will be recorded.

- *Participate in a Study:*
  Test users will attempt to discover some pre-created studies. They will then enroll in a study and complete it. The test user's experience with the study participation workflow will be recorded.

- *Remove Funds from Account:*
  Test users will remove funds from their account after being compensated from completing the previous study. The test user's experience with removing funds from their account will be recorded.

Regarding the number of tests, it would be beneficial to test out as many features of GeekSurvey as possible. Currently, this would involve having users test out the features related to the previously mentioned tasks. Again, the tasks would point users to do something, but would avoid giving them explicit step-by-step instructions.

As for how many subjects are to be selected, the GeekSurvey development team will reach out to five or more prospective end users with a relevant technical background. This would allow the GeekSurvey developers to collect a meaningful amount of feedback, while also being able to thoroughly engage with prospective end users throughout the usability testing process. It is also possible that the GeekSurvey development team will contact at least one of the clients to participate in the usability testing, as both GeekSurvey clients are software engineering researchers who publish large volumes of software engineering research.

The information and feedback gathered from the usability testing process will be recorded and gathered by having the GeekSurvey developers conduct interviews with prospective end users. These interviews will likely be conducted remotely through a video call. The GeekSurvey developers will likely have the end users share their computer screens and web browser as they use the GeekSurvey web application. The GeekSurvey developers will then have an interview form with a list of tasks they will have the end users attempt to complete. As the end users navigate GeekSurvey and complete tasks, the GeekSurvey developers will be observing and documenting their actions in the interview form.

The information that is recorded and gathered from the interview forms will be analyzed by having the GeekSurvey developers compare the interview forms. It is through these comparisons that any recurring themes or comments will be made apparent. Once these recurring themes and comments are noted, the GeekSurvey developers will make sure to document and prioritize the necessary changes as future user stories to improve GeekSurvey.

5. Conclusion

In conclusion, GeekSurvey is a web application that contains some elements that can be tested using unit testing, but should ideally be refactored to better accommodate for it. It interacts with external platforms to function as intended, hence why the team will need to test the integration of these platforms with the web app. Finally, the many features and functionalities of GeekSurvey will need to be tested with multiple end users interacting with one another to determine if any improvements can be made or if anything is broken. These tests will make sure that GeekSurvey is ready to be presented and handed off to the clients and future developers. As of March 2022, all of the required features for GeekSurvey have been implemented and the user interface has been improved. The team is on track to deliver a fully functional product by the end of April and even improve the current features.