**Software Dependability and Security**

Developers Inc.

**Assignment # 4**

**Comp3520**

Jagjit Singh Bilkhu - T00533766

Fahad Murtaza - T00527466

TABLE OF CONTENTS:

[**Executive Summary**](#_3xvgjqrn7xq1) **3**

[**Introduction:**](#_rmany64avs6e) **4**

[Platforms and Frameworks:](#_gyt700fqrs1h) 5

[Operational Verification & Security:](#_aodoudm992uv) 6

[**Conclusion:**](#_518v4m7uvbca) **8**

# 

# Executive Summary

Developers Inc. our company focuses on developing and delivering native mobile applications to provide cost-effective and robust solutions to the needs of our customers from different business fields. To develop and implement the solutions, we focus on a combination of scrum agile development with bits of extreme agile programming. In our current project we are developing a native mobile app for a restaurant business to display its menu and prices for each item. The app would be used by in house staff to take orders from customer tables and send them to the kitchen where the chefs would be able to know and prepare them. It will also allow tracking time to prepare the table order, accept payment from the customers at the table, track inventory and time clocks for the employees.

This report covers an analysis of our development process to verify secure design, and specify the operational requirements that the client needs to follow to ensure the dependability of our software. Resilience of our system will be considered in the context of how the system can be brought to function again in case an issue arises that compromises the dependability of the system.

# 

# Introduction:

We use an agile development process in our organization. This helps us to break the software system in small chunks and work on them iteratively in the implementation and development stages.

Our process is flexible to allow and accommodate changes in customer requirements and features because of this iterative nature. However, our process avoids having to do extensive and detailed documentation thus not making it a reliable and dependable in case there are major and fatal feature requirements by clients. Also, since our application doesn't include extensive verification of requirements, the chances of misunderstanding them is high and may result in a product function that is not required by the customer.

Another issue with our process may be that since we focus mainly on the skills of people, it will be hard to get replacements for team members in case anyone quits or falls sick. This may cause the project to drastically slow down and may even bring the progress to a halt at any stage in development.

In order to reduce the probability that such misunderstandings, during our sprints we have regular and recursive validations done with our clients to ensure that the product progress is occurring in the desired direction. Also, as new features are implemented, regular tests will be run on the system to ensure that the new additions do not affect the previous functionality.

To save time and to create a reliable system, we will make use of some third party platforms and tools such as Firebase and PayPal. This will help improve the product dependability and reliability since the third party tools have been already extensively used, tested and are more stable than when we have to implement them on our own. This further saves us time and will allow us to focus and make sure that the controllers and interface codes are much more stable and can be tested more.

To include third party tools and make our application more reliable, we decided to make some major changes in our development process that are listed as follows:

1. Instead of implementing the database servers from scratch, we decided to use readily available cloud databases such as Firebase and AWS.

For our restaurant application, we would use the **Firebase** platform that offers already implemented **Authentication services, Real time database, Messaging and Crash Analytics** that have been extensively tested and verified. This would be a great time and cost saver since it would be a lot of effort that would be required to implement these features and ensure that they are dependable and up to standards of required.

1. Another change would be creating multiple interfaces/modules: one specific for admins, and another for kitchen staff and servers.

The Admin module would have functions to add/edit/remove menu items, update prices, view employee logs, add/remove employee users and view employee hours. This module would be web based.

The employee module would be the one used by servers and kitchen staff to take table orders, collect payment, manage/update inventory, clock in/out, view order history and mark.

1. Payments would be collected using NFC via platforms and support from PayPal, Zooz and Google Payment Solutions. This would ensure the safety of the data transactions and also validate customer data.

To make our process result in more dependable and reliable products, we will make use of analytics tools that will report any errors and collapses that occur in the production environment. This will help us have a wider access to ad hoc tests and we can identify more test cases and scenarios that may have not been created when processing the requirements.

### Platforms and Frameworks:

Once initial requirements have been received from the clients, we will begin finding and identifying platforms/frameworks that would be able to provide the required functionalities. Based on the application type that is required by the client, the appropriate service levels of the platforms would be purchased and verified that they are able to satisfy the needs of the system.

To save cost and ensure that the selected platforms have sufficient space for all users, the operational tiers(pay scales) will be adjusted such that the client is able to save money as well as provide service to as many users as possible. This will be done by tracking the number of users which is provided by analytics tools integrated in the platforms.

In case a platform becomes unavailable or shuts down for example the shutting down of PARSE mBaas was a big shock to many software systems, we will ensure that the platforms chosen are exportable and may be used with other platforms. Since some platforms are not compatible with each other, we will then have to reimplement the functionalities with other available platforms. This is one major risk and disadvantage that we may face with third party tools and having a non dependable process.

### Operational Verification & Security:

For security, we will include more client side input verification and validation to ensure that the users enter valid data for example: Not allowing negatives in price values of menu items: This would prevent any further errors that may occur with the pricing. To avoid user errors, we the staff module would have very little input taken from user for example, the main requirement for input would be to login, or enter client card data. However, for the Admin module, the major validations would be done on permissions and pricing. This is because any change made by Admin would be affecting all staff module users. Therefore the credibility of the staff module would depend on the Admin module.

To avoid unauthorised access, the admin would be able to add or remove staff users and the Admin account would only be provided to the manager or owner of the client restaurant. Also, each restaurant would be provided with unique keys that will be used to store data in the database. These keys would be generated once the restaurant purchases the application from us to avoid interference in restaurant data.

During our testing phase, we will provide a beta version of the application to clients to test, verify and validate to ensure that they are understanding and comfortable with the functioning of the Admin and Staff.

Also, as more restaurants begin using the application system, regular checks to ensure that the database and users do not exceed the required numbers will be made. This is to ensure that the application is always readily available for all restaurants to use. Incase the usage exceeds the tier restrictions of platforms, the tier would be increased allow for more users and more data storage.

To avoid operational errors and external attacks, we would make recommendations to the client such as using private VPN to allow connectivity with the internet and not using public access networks. Also, use Auth Sessions to automatically logout a user in case of inactivity for a certain period of time.

Also, we will provide with the admin with ability to restrict/block certain staff devices incase they were stolen or are inactive.

To allow the admin to monitor all staff actions and activity, we will allow applications to log and report the staff operations.

Resilience and Error Recovery:

Using version control systems in our development process will help us recover incase our front end application or backend crashes or gets filled with incorrect data.

Also, to detect errors and crashes that occur in the production environment, we will make use of tools such as Google Analytics. These analytics will provide us with the error logs and will help us identify which kind of events caused them. In this way, we can quickly identify the errors and try to implement the solution as soon as possible. These analytics would also help in the long run when the software enters the maintenance stages of development to fix minor bugs that exist.

Since we are using a third party platform as a backend i.e Firebase, the probability that it may crash or have internal errors is very low. However, in case an error was to occur, we would redeploy the system as soon as possible to ensure that there is minimal downtime.

# Conclusion:

In summary, we will change our development process to include use and testing of third party tools such as Firebase instead of creating and hosting all on our own. This will avoid having to implement these tools on our own and also reduce the testing and maintaining time spent on them, Instead we would focus greatly on making the applications front ends and controllers more dependable, reliable and resilient by performing extensive tests on them.

Reference

UsersNap- “A guide to Software Resilience testing” - Feb 15, 2017 <https://usersnap.com/blog/resilience-testing/>

Wikipedia - ‘Cloud Database’ - Development models - 4th december 2017

<https://en.wikipedia.org/wiki/Cloud_database>

UXmatters - "Reliability and Dependability in Usability Testing” - june 25 2011

<https://www.uxmatters.com/mt/archives/2011/06/reliability-and-dependability-in-usability-testing.php>