SOFTWARE PROJECT PLAN

1.0 Introduction

The student Internship management system is an Android app that is an online internship management android application in which a student can be able to apply for an internship by submitting his application form online through the application to the concerned faculty on the other end faculty could be able to go through the application form that is submitted by the student and review the details of the employer during which the student can be able to find the internship. The admin can examine both the overall number of students and the number of students who applied for internships in the meanwhile.

The development of the software application will result in the paperless, real-time streamlining of the application process, which might help the company save time. Many students will find it helpful to apply for an internship using this straightforward software since it will be the only place, they can seek rather than going via several job portals and applications and monitoring them, which will be quite challenging. Many students also are unaware of the application procedure for internships, which causes them to lose out on possibilities. Our application will serve as a central location for students from various universities to apply for internship opportunities in any department of any organization.

1.1 Project scope

The scope of the system is to provide a friendly Android Application which is easy to navigate and at the same time provides user friendly information about the system and how it works.

The scopes of the project are:
• Android application.
• Users:
  o Students of any department.
  o Faculty Supervisors.
  o System Administrators.

1.2 Major software functions

There are 4 major functions namely input, Processing, Storage and Output.

1.3 Performance/Behavior issues

- Software Fragmentation.
- Device Fragmentation.
- Testing Fragmentation.
- User Interface Design Rules.
- Programming Languages.
- Integration of third-party APIs.
- Security Issues.
- App Visibility.

1.4 Management and technical constraints

Since there are not much hardware components connected in the project that might be good and coming to the fire base for the cloud that might be a technical constraint while purchasing the subscription, since the application is developed in the Android studios in java languages that comes down to the java libraries where we might be working with the limited sources of packages.

2.0 Risk Management

In order to reduce, monitor, and control the likelihood or impact of unpleasant events or to optimize the realization of possibilities, risk management involves the identification, evaluation, and prioritizing of risks. This is followed by the coordinated and efficient use of resources.

2.1 Project Risks

The following are the Project Risks: -

1. Customer related Risks.
2. Communication related Risks.
3. Market Risks.
4. Resource risks.
5. Financial Risks.
6. Technical Risks.
7. Managerial Risk.
8. Performance Risks.
9. Maintenance Risks.
10. External Risks.

2.2 Risk Table

The complete risk table is presented. Name of risk, probability, impact and RM3 pointer are provided.

S. No	Name of risk	Probability	Impact	RM3 pointer are Provided
1	Customer related Risks	10%	Mostly Occurred	Requirement analysis is the most important task in design team or developer. related to requirement analysis risk. Specification related Risks: Customer plays crucial role requirements need a proper understanding and the client needs get misinterpreted, and the product obtained is not as per the requirements of the customer. Improper analysis of the customer requirements results into
2	Communication related Risks	10%	Mostly Occurred	There needs to be a streamline communication between the client and developer to come up with the software product that meets up all the needs. otherwise, it would lead to fatal design errors. Design errors

S. No	Name of risk	Probability	Impact	RM3 pointer are Provided
3	Market Risks	10%	Mostly Occurred	Prior to the implementation of new mobile software, a in depth research of the market needs to be carried out. also, important to come up with something new or
4	Resource risks	20%	Will be a bit not sure	The dependencies encountered in developing a mobile application
5	Financial Risks	0%	No	Budget and cost estimation plays a vital role in the project about the estimated expenses and other costs that can be incurred in the manufacturing of the software but in my project is a team project.
6	Technical Risks	90%	Major impact overall performance	A correct application design is half the work done. But a faulty application design consumes a lot of time and money and often proves to be risky in the long run.
7	Managerial Risk	40%	Regulated	In this highly competitive software evolving market first to release their product first in market so they can attract customers easily and acquire market share from very Mobile platform updates are rolled out rapidly, mobile applications should be updated simultaneously to considered and software is exposed to instability and
8	Performance Risks	70%	High impact	The security attacks from the external environment. platforms such as check on the software vulnerability and methodologies to Enhanced security check algorithms must be employed within the software application to keep an eye on the security attacks. algorithms can be proved helpful in securing a system. depend on platform and development process and that are.
9	Maintenance Risks	20%	Very according to the data	launch of the application in the market. Providing support Regular updates in platforms are rolled out so application should adopt new changes in platform and provide support be updated regularly.
10	External Risks	10%	Yes virus	Many applications in the market need certain requirements Certain applications Proper testing should be done in for example, if an application is internet dependent, then we should test it on slow as well as fast connections to ensure that the application does not.

2.3 Overview of Risk Mitigation, Monitoring, Management (RM3)

2.3.1 Customer related Risks

Requirement analysis is the most important task in developing a mobile application because it includes an in-depth market research along with client interaction. Risk factor is very high if any requirement is misinterpreted by design team or developer. Below is broad description related to requirement analysis risk. Specification related Risks: Customer plays a crucial role in any software engineering process. Customer requirements need a proper understanding and documentation prior to the advent of the project development. Many times, it so happens that the customer needs are not clearly specified due to lack of technical knowledge of the customer. There have been cases when the client needs get misinterpreted, and the product obtained is not as per the requirements of the customer. Improper analysis of the customer requirements results into loss of time and affects the quality of software delivered. It has been noticed that frequent changes in the software design led to buggy effect. Sometimes the customer demands are unreasonable (e.g., speedy development to get the first mover advantage in a particular feature), this might lead to poor quality and cause a risk to the overall project

2.3.2 Communication related Risks

There needs to be a streamline communication between the client and developer to come up with the software product that meets up all the needs. The developer should have a clear picture of the product and his targeted audience otherwise it would lead to fatal design errors. Design errors in mobile software often prove to be very risky and expensive on the long run. To avoid this type of risk we should invest adequate time in explaining the vision of the application, leading to greater enthusiasm and greater team coordination.

2.3.3 Market Risks

Prior to the implementation of new mobile software, a in depth research of the market needs to be carried out. It is risky to come out with a software product that doesn't not cater to the market needs or suffice the customer needs. It's also important to come up with something new or something that would add value to existing available options.

The research should be based on the following criteria: Is there an existing market or new market needs to be created? What are the competitive products present in the market? What is the purchasing power of the customer? Any specific issues related to the user interface.

2.3.4 Resource risks

The dependencies encountered in developing a mobile application can be of the following types:

• Resource dependency: The mobile application is sometimes dependent on other shared resources of the memory, servers and skilled staff. The availability of those resources only facilitates in proper functioning of the application.

• Platform dependency: When an application is built, it is dependent on features provided by the platform and should not breach platform security. For example, certain features are supported in one platform while not on another (e.g. music files can't be shared in iOS while in other platform this can be done). To avoid this, we should focus on developing applications in which the GUI is separate from the core logic, so that we need only change the GUI part for different OS and the logic parts remains the same.

• Stakeholders Investment: The development and launch of a mobile application largely depends upon the support gathered from the stakeholders. Stakeholders play a crucial role in finance and the budgeting and promotion of the product. The amount invested and the policies used for campaigning of the mobile application are of prime importance.

2.3.5 Financial Risks

Budget and cost estimation plays a vital role in the development of any mobile software application. It should be made clear with the clients before taking over the project about the estimated expenses and other costs that can be incurred in the manufacturing of the software. Buffer amount is always appreciated as it would decrease risk of financial crisis during multiple stages of the project and ultimately increases profit at the end.

2.3.6 Technical Risks

A correct application design is half the work done. But a faulty application design consumes a lot of time and money and often proves to be risky in the long run. Application design includes the following:

• Algorithm: The algorithm used for the application should be cost optimal as well as space optimal. The application should not occupy a lot of memory in the system.

• Platform: The platform on which the application is going to run should be feasible in nature and should be one that is relatively stable and secure. The most popular mobile software application platforms are Android, iOS and Windows.

• Graphical User Interface (GUI): The application GUI proves to be major factor in attracting more audience for a mobile application. A complex unappealing GUI can be risky as it fails in providing ease of usage to the customers. The application GUI should be user friendly along with certain enhanced features which make it stand apart from the rest of the applications already available in the market.

• Testing Strategies: There are various testing strategies that are available to test the proper functioning of mobile software. To be on safe side and as a precaution different testing strategies should be employed. The faults encountered in mobile software after the device is launched in the market are difficult and expensive to tackle. In general test cases should be prepared for essential functionality. The application should be tested on various available models so proper functioning is done on all models as same platform has various independent models (i.e., various dimensions a particular application should be able to work on devices with varied screen size and resolution as well as varied vendors). Network related testing should also be done as all networks have some change in their protocols. For example, an application might give unsatisfactory behavior on slow speeds (2G).

In order to stay safely away from such risks, it is suggested to carry out testing on field testing on field includes giving the beta version of the application to the clients for usage for a pre-determined time period and taking their reviews of the product. The suggestions and feedbacks of the users should be carefully noted down and taken into consideration. Another way of testing is to perform Stress and Load testing, these help us to avoid DOS attacks (due to excessive loads), some crashes that occur in unthought-of of scenarios, and help us to point out whether there is any memory leak or any performances issue causing the device to slow up or hang.

2.3.7 Managerial Risk

In this highly competitive software evolving market first mover always has extra benefits. Every organization wishes to release their product first in market so they can attract customers easily and acquire market share from very beginning. Even there are deadlines to be followed for projects. Mobile platform updates are rolled out rapidly, mobile applications should be updated simultaneously to improve efficiency and proper utilization of platform features. Due to pressure, various scenarios are not considered, and software is exposed to instability and security risks.

2.3.8 Performance Risks

A good application is one which can deal with the security attacks from the external environment. High risks are involved in running insecure mobile software and should be handled with topmost priority. Different platforms such as Android, iOS, RIM, etc. have their own ways on keeping a check on the software vulnerability and methodologies to deal with external threats. Enhanced security check algorithms must be employed within the software application to keep an eye on the security attacks.

2.3.9 Maintenance Risks

The job of a developer does not get over with just the launch of the application in the market. Providing support after release is most important thing for product success. Regular updates in platforms are rolled out so application should adopt new changes in platform and provide support for that. There are risks associated after the launch that must be dealt on timely basis. The features of the software need to be updated regularly. Clients often have customized needs requiring constant updates and needful attention.

2.3.10 External Risks

Many applications in the market need certain requirements to be met for their proper functioning. Certain applications use GPS services while most of them require internet facility for their running. Proper testing should be done in all types of environments. For example, if an application is internet dependent, then we should test it on slow as well as fast connections to ensure that the application does not either hang / crash or cause any unwarranted behavior in any situation. Also, we should take into account what would happen if the data connection dropped all of a sudden and the amount of battery usage under the above-mentioned conditions.

The above stated situations are just some scenarios in which one should test and minimize the risk factor for application.

3.0 Project Schedule

As this is a big project and it need a lot of help with others in areas of fields, so we are going to set a repository in GitHub.

3.1 Project task set

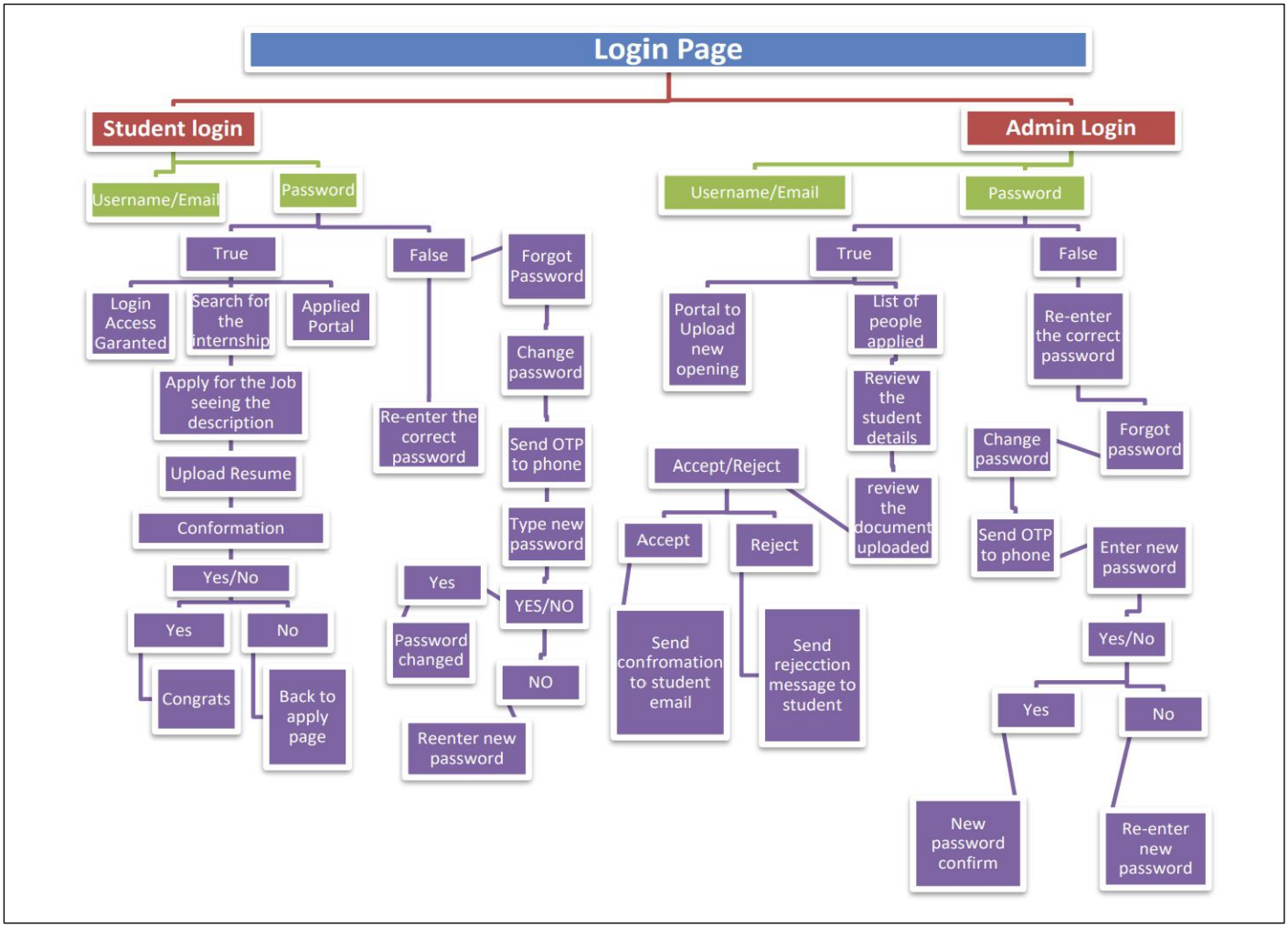
In this project we are going to use the following project task set as we start working on the project.

- Waterfall model
- V model
- Incremental model
- RAD model
- Agile model
- Iterative model
- Prototype model
- Spiral model

3.2 Functional decomposition

Start date of the project: - 10/5/22
End date of the project: - 12/5/22
Documentation: - Gangula Harshini (2845690)
Resources: - Nischal Gudehindler Lingeswara (2825960)
Coding: - Savan Yeshwanth Rao (2784780)
Testing: - Vadde Krishnasai (2805416)
Overall working to gather but we will for now divide the work according to the theory part.

3.3 Task network



3.4 Timeline chart

We will be following the TimeLogCarbon.txt format as referred in the black board.

3.5 Schedule compliance

We will be reporting the Continually track, review adjust and report on the project's performances accordingly to the GitHub and in the Grail.

4.0 Appendix

https://github.com/vaddekrishn/CIS-634