# OVERVIEW

## Introduction

*Tournament Organizer* is an android project for tournament organizers. Any level of tournament organization would be eased with *Tournament Organizer*. This project intends to digitalize tournament organizations. This a completely new project which is going to be built from the scratch. The budget category for this project is low. This project will take from 3-4 months. *Tournament Organizer* is going to be developed by Scandium Technology Private Limited. This project has no direct or indirect effect on any other projects. Other projects will not have any contribution in this project.

*Tournament Organizer* is a new product for the market. It uses technology for what the organizers are using hardcopy sheets. This project is intended to gain minimum profit with maximum service.

## Background

In a recent sports tournament where many people participated, the organizers mentioned the problem they faced during the tournament. They particularly stressed on the hard work they had to do in organization of data with traditional way involving pen and paper. Scandium Technology Private Limited made note of the situation and planned to make their work easier. We believe that the task stress can be heavily reduced with use of technology. Then, we decided to make an application for the ease of organizers.

## Description of the proposed system

The system includes many features directed to fulfill its aims and objectives.

### Features of the project

Due to time limitations the project features were limited to:

* Create new user accounts (separate for organizers and participants)
* Save user details in remote database
* Edit user details
* Organizers can create new tournaments or modify current tournament
* Screenshot of the app can be taken for ease of sharing
* Participants can participate in created tournaments
* Winner predictor with AI
* Both league and cup format tournaments can be created
* Custom tournaments with custom format can be created
* Save tournament data in remote database

# SCOPE

Scope can be called the limitation of the project. It is the extent to which something is affected due to a certain action. Project Scope is the extent to which the project can affect other areas.

## Aims of the project

The main aim of the *Tournament Organizer* is to facilitate tournament organizers and participants. This project aims to make it easy for anyone involved in any sort of tournament competitions. It aims to remove any sort of hard work of maintaining tournament data and information that tournament organizers need to do. It aims to keep the participants of tournament well informed than they previously used to be.

## Objectives of the project

Below are the objectives of the project:

* **Establish a remote server** to facilitate storing of tournament and user information.
* **To perform survey** of organizers and participants
* **Allow user permissions** to register and edit their accounts
* **Permissions for organizer** accounts to create, edit and terminate tournaments
* **Business goal** - finishing the project under 10,000 NRs and within 4 months.
* **Interaction between servers and client devices** need to use minimum resource and be very quick.
* **Test the application** in various versions of android
* **Clean complete code** so that any changes needed can be done easily
* **Bug free application** so that users are not disappointed with the application
* **Provide clean UI and UX** to make the application easy and interactive
* **Provide no harm** to social, cultural and environmental aspects with the application
* **Use latest architecture and standards** for application quality

## Overview of Scope of project

The project has certain limitations of its own. The project is specially targeted for people involved with sports and other various types of tournaments or indoor and outdoor games. However other users may use it if it helps them in any way. Most of the details (ex: goal scorers in a football match, assists) can be recorded however storing each detail (ex: complete passes, dribbles in football, distance of six struck in cricket) cannot be stored because of time constraints. Any help may be provided to users through communication. User queries can be responded to and bugs will be fixed. The application is limited to android devices Android 4.0 only.

# DEVELOPMENT METHODOLOGY

**Agile Development Methodology**

*Tournament Organizer* is going to be developed using Agile Software Development Methodology.

Agile Development Methodology is iterative development method. Usually, agile development method involves creation of different teams who collaborate to continuously evaluate/change requirements and solutions. The teams are self-organizing and cross-functional. Scrum is a subset of agile.

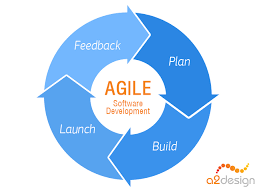


Figure 1 Agile development methodology

Agile development methodology has many advantages. Some of them are:

* **Stakeholder Engagement**

Agile is full of opportunities for engagement between stakeholder and team. This increases chances of collaboration between the client and project team.

* **Transparency**

Clients can see progress of work in agile development methodology. They can review software builds and express their opinions.

* **Allows for change**

There is always an opportunity for constantly refining and repriortizing the overall product. This allows changes in short period of time.

* **Predictable costs and schedule**

The cost is predictable as each Sprint is a fixed duration and the work done by a team in a fixed time box can be limited. It improves decision making about features.

* **Focuses on users**

Agile focuses on acceptance criteria to define product features. This provides opportunity to beta test software after each Sprint, which provides ability to make changes as needed.

Agile development has some disadvantages along with its advantages. Some of them are:

* **Active user development throughout**

As agile requires user participation, they need to be prepared regularly which is very engaging. It requires big commitment of user during the project development.

* **Requirements change**

Continuous change in requirements can cause ever-lasting projects. There is less predictability at the start of project and end product. It makes is hard to negotiate and fix the price.

* **Testing throughout the life cycle**

Testers are needed throughout the project and this increases the cost.

* **Frequent product delivery**

The stakeholders need to be ready and available for prompt testing of features. This can be time-consuming.

Although agile development has some disadvantages, I believe that its advantage overwhelm the disadvantages in case of my proposed project. The requirements of *Tournament Organizer* may change during development. Continuous tests need to be performed during the implementation phase. So I chose agile development methodology for its benefits on my project.

**Model-View-Controller design pattern**

The model-view-controller design pattern consists of data model, control information and presentation information. These three are separated from each other. The model contains application data, view presents the data to user and controller listens to events triggered by view and model.

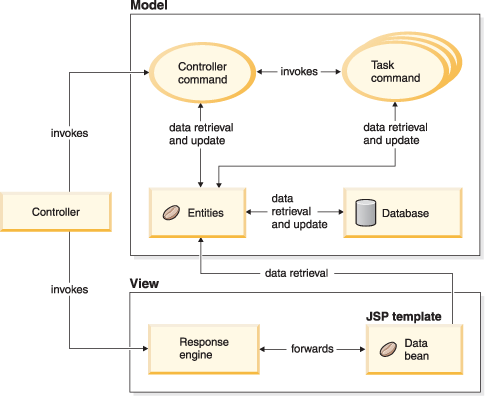


Figure 2 MVC design pattern

Model-View-Controller design pattern has many advantages. Some of them are:

* **Faster development process**

MVC supports rapid and parallel development. It is approximately three times faster than other design patterns.

* **Modification doesn’t affect the entire model**

Model is not dependent on other parts so any changes in model will not affect the entire architecture.

* **Returns data without formatting**

MVC returns data without any sort of formatting so same components can be called for use with any interface.

* **Ability to provide multiple views**

In MVC, we can create multiple views for model.

* **Support for asynchronous technique**

MVC produces applications that load very fast.

MVC has some disadvantages too.

* **Increased complexity**

MVC division of model, view and controller increases complexity in small programs.

* **Need of multiple programmer**

More programmers are needed in MVC as each needs to specialize in one of its parts.

* **Knowledge on multiple technologies is required**

MVC requires programmers to have knowledge of the design pattern along with other technologies.

* **Inefficiency of data access in view**

As model, controller and view are separated from each other there may not be data access in view which could sometimes make our work easier.

Model View Controller is the most preferred design pattern for building android applications. The structure of android programming and the vast advantages of MVC design pattern for my project has encouraged me to use MVC design pattern for this project.

**Object-oriented method**

Object-oriented method combines data and methods into objects. Objects correspond to real things a system deals with. Complex relationships are represented in an easy manner in Object-oriented method. WE focus more on the behavior of system in object-oriented method. Class or Object is the main feature of object-oriented programming.

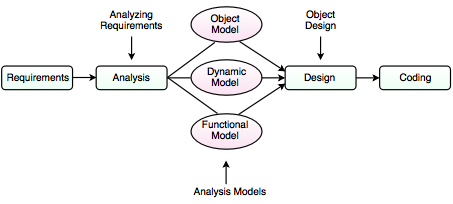


Figure 3 Object oriented method

Object-oriented method has several advantages:

* **Code reuse and recycling**

Objects created in object oriented programs can be easily reused in other programs.

* **Design benefits**

Object oriented programs force designers for extensive planning phase which reduces flaws. Once a program reaches certain stage, object oriented programs are easier to write.

* **Software maintenance**

Object oriented programs are easier to modify and maintain. So a lot less work is needed to maintain object oriented programs.

* **Encapsulation**

Knowledge of implementation is no necessary for use of an object. They can be used according to their functionality.

Some of the disadvantages of object oriented method are:

* **Large program size**

Object oriented programs usually contain more lines of code than procedural programs.

* **Steep learning curve**

It may take time for people to get used to object-oriented programs. Key techniques like inheritance, polymorphism may be hard to learn at the start.

* **Slower programs**

Object oriented programs tend to be slower than procedural programs since more instructions need to be executed for similar tasks.

Object oriented method is the modern method of programming. *Tournament Organizer* deals with lots of accounts, stages etc. and these entities can be dealt as objects. So, I preferred object oriented method for my project as my project would be a lot easier through this method and don’t need to be much concerned about resources available.

# PROJECT PLANNING

## Work Breakdown Structure

Work breakdown structure is the process of dividing complex projects to simpler manageable tasks. It is not restricted to single field and can be used for any type of project management.

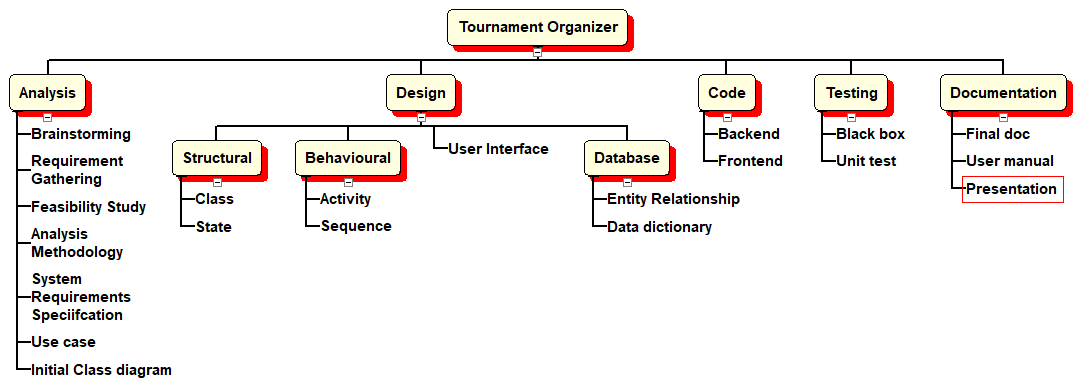


Figure Work Breakdown Structure

*Tournament Organizer* has 5 main tasks namely: Analysis, Design, Code, Testing and Documentation. Various tasks are carried out during the analysis phase to analyze our application requirements. Brainstorming on the project features, analyzing requirements, drawing use case etc. is done in analysis phase. Different diagrams are drawn in design phase. Structural diagrams like class diagram, behavioral diagrams like activity, and database diagrams like ER diagram are drawn in Design phase. Frontend and backend code is completed after that. Then the application is tested with unit, black box and other tests. Finally, it is documented along with user manual and is presented to the client.

## Time estimation

|  |  |  |
| --- | --- | --- |
| **Task No** | **Task Description** | **Time** |
| 1 | Analysis   * Brainstorming * Requirements Gathering * Feasibility study * Analysis methodology * System requirements specification * Use case * Initial class diagram | 16 days   * 1 day * 3 days * 2 days * 3 days * 4 days * 1 day * 2 days |
| 2 | Design   * Structural * Behavioral * User Interface * Database | 35 days   * 7 days * 14 days * 7 days * 7 days |
| 3 | Code   * Frontend * Backend | 24 days   * 9 days * 15 days |
| 4 | Testing   * Black box * Unit test | 15 days   * 7 days * 8 days |
| 5 | Documentation   * Final document * User manual * Presentation | 20 days   * 10 days * 4 days * 6 days |
|  | Total | 110 days |

## Milestones

|  |  |  |
| --- | --- | --- |
| **Task No** | **Task description** | **Deadline** |
| 1 | Analysis   * Brainstorming * Requirements Gathering * Feasibility study * Analysis methodology * System requirements specification * Use case * Initial class diagram | 14th May   * 30th April * 2nd May * 4th May * 7th May * 12th May * 13th May * 14th May |
| 2 | Design   * Structural * Behavioral * User Interface * Database | 19th June   * 21st May * 4th June * 12th June * 19th June |
| 3 | Code   * Frontend * Backend | 12th July   * 28th June * 12th July |
| 4 | Testing   * Black box * Unit test | 27th July   * 20th July * 27th July |
| 5 | Documentation   * Final document * User manual * Presentation | 17th August   * 8th August * 12th August * 17th August |

## Scheduling

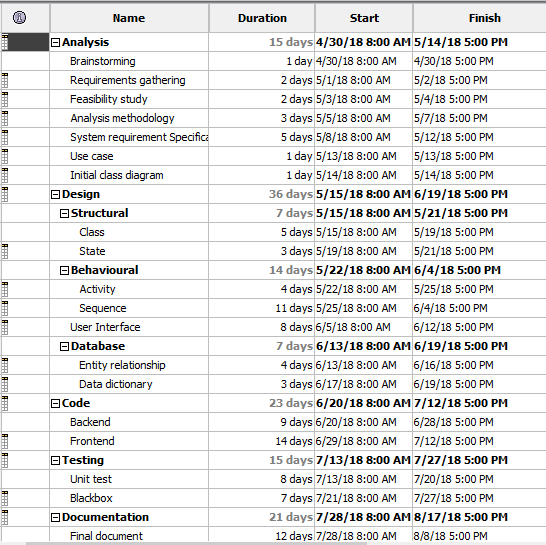


Figure 5 Gantt chart activities

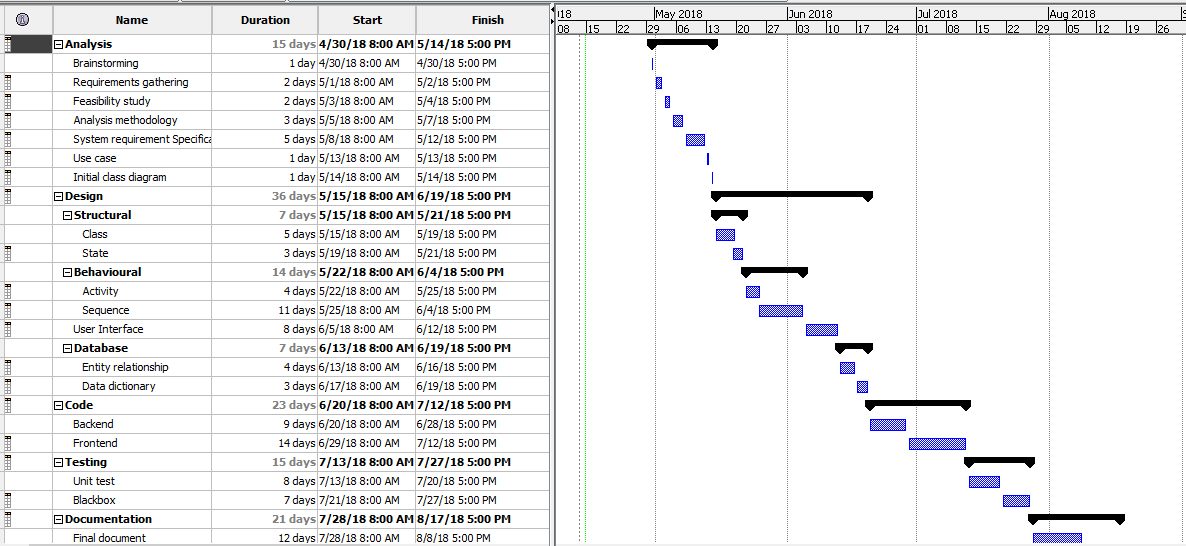


Figure 6 Gantt chart

# OTHER PROJECT ACTIVITIES

## Risk Management

To identify, analyze and respond to risks that arrives over lifecycle of project is risk management. Risk management should be part of planning process to find out the risks that might happen in project and control the risk if it occurs.

Risks could negatively affect project’s timeline, performance or budget. They become issues in projects. So, we need to identify, categorize, prioritize and plan for risks before they affect our project.

There are five steps of risk management:

1. Identify the risk

The team recognizes risks to the project

1. Analyze the risk

Find the consequences of the risk and its nature

1. Evaluate the risk

Find the magnitude of risk with likelihood and consequence

1. Treat the risk

Find high impact risks and plan to modify the risks for opportunities

1. Monitor the risk

Review the risks and monitor and track them

Following table helps us analyze risks

|  |  |
| --- | --- |
| **Consequences** | **Value** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

The probable risks in *Tournament Organizer* are:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Risk** | **Consequences** | **Likelihood** | **Impact** | **Action** |
| 1 | Developers leaving job | 3 | 2 | 6 | Hire new developers |
| 2 | Lack of budget | 4 | 1 | 4 | Increase project budget after consulting with stakeholders |
| 3 | Stakeholder conflict | 5 | 1 | 5 | Remind them of project requirements signing at the start |
| 4 | Scope is ill defined | 2 | 2 | 4 | Redefine scope |
| 5 | Communication overhead | 3 | 1 | 3 | Remind stakeholders of project state |
| 6 | Infeasible design | 2 | 2 | 4 | Hire designers |
| 7 | Technology is not compliant with standards | 2 | 1 | 2 | Change technology to modern one |
| 8 | Failure to integrate with business process | 4 | 1 | 4 | Integrate business process |
| 9 | Project team lack authority | 4 | 2 | 8 | Provide proper instructions and manage team |
| 10 | Business change impact project | 3 | 1 | 3 | Change project requirements according to business change in next sprint |
| 11 | User interface is low quality | 4 | 2 | 8 | Change user interface |
| 12 | Users reject the project | 5 | 2 | 10 | Gather requirements again and perform required changes |
| 13 | Project doesn’t match organization’s culture | 3 | 1 | 3 | Adjust organization’s culture for the project |

If the developers leave the job new developers can be hired. Or the leaving ones may be persuaded to stay. If new developers are hired they can start in next sprint. If there is no proper authority in the project, developers can be reminded of their duties. New manager can be appointed or brought in. User interfaces can be changed in next sprint. There is a medium chance of user interface being bad. If the users reject the project consult with them and find the lacks. Then work on fixing the issues in next sprint.

## Configuration management