Warhammer 40K Unit Efficiency Calculator  
Technical Document

**Group Members:  
Frederick Zuberer (**[**fzuberer@kent.edu**](mailto:fzuberer@kent.edu)**)  
Dan Maher (**[**dmaher2@kent.edu**](mailto:dmaher2@kent.edu)**)  
Anthony Telerico (**[**ateleric@kent.edu**](mailto:ateleric@kent.edu)**)  
Winson Li (**[**wli21@kent.edu**](mailto:wli21@kent.edu)**)**

**Software Engineering Project  
Kent State University**

**Table of Contents**

**Table of Contents [ii](#_gjdgxs)**

**Revision History [ii](#_2s8eyo1)**

**1. Problem Statement [1](#_17dp8vu)**

1.1 The Problem to Solve [1](#_3rdcrjn)

1.2 Who Would Be the Customers? 2

1.3 Who are the End Users 2

1.4 Constraints Imposed by the Customers 2

**2. Elicitation Plan, Assumptions, and Risks 3**

2.1 Requirements Elicitation 3

2.2 Assumptions and Risks 5

2.3 Usage Scenarios 6

**3. Application Design and Implementation 6**

3.1 User Interfaces 6

3.2 Look and Feel Guidelines 7

3.3 Define Menu Hierarchy and Traversal 8

3.4 Define Screens and Content 9

**4. Development 9**

4.1 Define Reports and Content 9

4.2 Define Documentation and Training Guidelines 11

**5. Other Requirements 12**

5.1 Application Functionality 12

5.2 System Architecture/Software Design 12

5.3 Error Recovery 12

**6. Conclusion 13**

6.2 Summary/Conclusion/Future Works 13

6.3 Acknowledgements 13

# Problem Statement

## The Problem to Solve

The problem we are tackling to solve relates to the mechanics of the table-top wargame known as *Warhammer 40,000* or simply *40k*. *Warhammer 40,000* is the most popular miniature tabletop wargame in the world. The game uses model soldiers, tanks, aircraft, and terrain to construct a ‘battlefield’ in which players command their forces and seek victory. To facilitate gameplay the system is based on the use of six-sided dice which creates a game based upon random probability and statistical outcomes. Amongst the community for this game is a concept known as “Mathhammer” in which the players who have a knowledge of statistics calculate the various expected average outcomes of attacks made by one unit against another.

Our aim is to create a simple calculator that considers the units various stats, any bonuses or determinants and the enemies stats that can negate any damage done. The process for making these attacks is dependent on a pool of six-sided dice that are rolled to generate successful hits, then the number of successes is used to generate how many of those hits effectively wound the target and then finally those wounds are attempted to be resisted by the units’ saving stats.

With all the information input into the calculator by the user and the calculation run would output the statistical average damage dealt for that particular set of dice rolls. This information is valuable as playing the averages is how games of chance are won, and this being a diced based game at the end of the day is a game of chance.

As you can see the process is a simple series of dice rolls. However, the best players know how to use their units well, and what their capabilities are. Many new players and even some veteran one’s stumble with being able to truly understand a unit's effectiveness. To do so they need to know the statistical average damage output that can be expected from that unit on a variety of targets. This information can help them in many ways:

* Knowing the best role for their units.
* Understanding the threat posed by the opponent’s units to their own.
* Model kits are expensive for the game, knowing the capabilities of them before spending the money would be wise.
* Gaining a better understanding of the various effects of the bonuses and determinants placed on their own and foes units.

This calculator would be a simple tool able to be used by new and veteran players to better enjoy the hobby. We aim to include functionality that defines the key terms such as ‘Ballistic Skill’ and ‘Feel No Pain Save’ to help the new players grasp the concepts better as well. 40k is a fun hobby enjoyed worldwide, with a thriving community and even international tournament play. Our aim is to give that vast player base a tool that can improve on an already beloved game.

## Who Would the Customers Be?

We are working with Shieldwall Gaming Club to develop the application. They are our shareholders. The club has a large player base, active tournament leagues and weekly casual games of 40k. The club is a good representation of how many players participate in and play the game.   
  
The customers would be the 40k playerbase, who also would be the end users of this product. For a unifying name we can say “40k Playerbase''. The player base would want the calculator to be simple to use and account for all modifiers that can affect the dice rolled in the process of making an attack.

While the user base cannot be specifically determined due to the nature of its “table-top” format, a popular forum dedicated to the game called Dakka has over 100K registered users that actively play the game. YouTube is a popular site to find guides and dedicated Warhammer players with hundreds of thousands of subscribers and equally as many views.

## Who Are the End Users?

The “40k Player Base” would be the end users. The pool of end users is large, as this is a game enjoyed by players of every description except young children. The average player of the game is a man in their mid to late twenties, working an entry level job and simply looking to enjoy a bit of escapism in a game with friends or indulge a competitive itch. Most players use a phone-based application to draft ‘lists’ or rosters of units made using the game's rules they know how to use, and often play with a smart phone at hand.

## Constraints Imposed by Consumers

* Calculations need to be accurate (little to no margin of error to be accurate).
* Rule and stat changes impact the process of performing calculations, so this needs to be accounted for.
* The calculator is VERY specific so even minute changes must be accounted for. Changes within the overall rulebook of Warhammer 40k impact the calculator as values and statistics might change.
* The calculator needs to be efficient/simple to allow for fast calculations to be usable in tournaments (time-limits exist).

# Elicitation Plan, Assumptions, and Risks

## Requirements Elicitation

### System Functionality

**What will the system do?**

The system takes the input of the user which includes information on the various units stats, any bonuses or determinants and the enemies stats that can negate any damage done and calculate the statistical average damage output for one unit versus another.

**When will the system do it?**

Upon completion of user input and completion of the calculation.

**Are there several modes of operation?**

No, not at this time. There is the potential to expand the system to allow for multiple game types, which would constitute different modes of operation.

### Data

**For input and output, what should be the format of the data?**

All data will be input and output in integer form.

**Must any data be retained for any period of time?**

The data must be retained while the application is opened until a new calculation is needed by the user.

**Are there any design constraints?**

Calculations will need to be accurate and rounded appropriately. The application must work consistently and reliably without an internet connection.

### Interface

**Is input coming from any other systems?**

No, all input is native to the application and/or is input by the user.

**Is output going to any other systems?**

No, all output will not be going to any other system.

### Performance Are there constraints on execution speed, response time, or throughput?

Since tournament play is time limited, the calculations need to be fast enough to not delay gameplay. Any delays must be from user error only.

### End Users

**Who will use the system?**

The Warhammer 40K player base during casual games and competitions.

**Will there be several types of users?**

No, the only users of this application will be players of the game. It needs to be noted however that there are players that follow different rulesets so different versions of the calculator will need to be created to account for this sub-playerbase.

**What will be the skill level of each user?**

The skill level will vary, but the user must be familiar with the basic Warhammer 40k rules (and whichever ruleset they decide to follow).

### Usability and Human Factor What kind of prerequisite knowledge or training will the user need to have?

The user will need to have basic knowledge of the Warhammer 40k rules.

**How easy should it be for a user to understand and use the system?**

The system should be intuitive for any user of the system, and will be able to be used without any outside instruction.

### Reliability and Availability

**Should the system detect and isolate faults?**

Yes, the system should detect faults and return errors to the users.

**What is the needed uptime of the system?**

Since the system does not require an internet connection, uptime is not an issue.

**Will the application require internet access?**

The application will only require internet access for initial download and future updates.

### Maintainability

**When and in what ways might the system be changed in the future?**

Rule updates happen periodically, and the system must be changed quickly to keep up with these changes. Also, there are several older versions of the game that a small percentage of the user base might want added in the future.

**How easy should it be to add features to the system?**

Features will be easy to add, and require a small update on the user end. This is the only time an internet connection will be required, and should automatically update.

**How easy should it be to port (migrate) the system from one platform to another?**

The technology and program will be easy to port from Android to Apple. The issue will be getting the app approved for the Apple app store, which is notoriously difficult.

## Assumptions and Risks

With a table-top game like Warhammer 40K and with the sheer amount of units that exist within the game currently, the calculator serves to simplify the calculations when dealing damage. Many assumptions must be made as inputting every unit into the calculator would cause the app size to be too large and slow for an app whose purpose is to be fast and efficient. It is assumed that a player owns the core rulebook and a separate rule book that includes the player’s faction and units within the faction. It is also assumed that the player isn’t following an older edition where there are differences in the rules and stats that the app will not be designed for. There is room for expansion to these older editions provided the initial calculator is received well and the request made for that expansion. Our calculator is also meant to be a tool and is not a dictionary for the game itself. 

There are a few risks to consider but nothing of serious concern. While editions have changed, they do not change frequently enough that there could be an overlap with development. Updates to the game do release and will more than likely have little to no impact on this core algorithm to gameplay but that chance is always there.  
  
These assumptions are verified through talking with our shareholders and through our understanding of the rules and mechanics. They are also verified from live app testing. The main constraints are our familiarity with Java and Android Studios along with time constraints in implementing more features and editions than currently implemented.

## Usage Scenarios The main base use case is the user (the actor) using the application to find the average damage their unit can deal to a target. This is what helps them determine the unit’s efficiency. The only other use case would be for looking up the glossary of terms. The information on statistical average damage would help users find out their unit’s efficiency. This info can be used in a variety of ways and situations. For example, this information can help in a use scenario where the player is in a casual game or tournament to help decide their best move or position or to assess the enemy. It can also be used in a scenario where the user is shopping and looking for the most desirable faction and units.

# Application Design and Implementation

## User Interfaces

## Look and Feel Guidelines

3.1.1 Simple and Clean Look

The application needs to have a simple and clean look. The range of colors for the background, buttons, text, etc. must be kept to a minimum and it should not make text or buttons hard to see or read. A Light Mode and Dark Mode would help the look.

3.1.2 Intuitive and Easy to Use and Understand

The application needs to be intuitive and easy to use and understand by even beginner Warhammer 40k players. The navigation of the screens should be easy. What is being asked for each input and what the resulting output is should be clear. A dedicated screen to view definitions and a question mark (?) button in the calculator screen that provides quick definitions would help newer users of 40k. Also, the built-in Android screen/text reader in Accessibility Settings is to be compatible. Error or incorrect input handling should also be there.

3.1.3 Fast and Efficient

The selection of the options and the inputting of the unit stats and necessary info should be fast and efficient. A Reset All has been implemented to better allow fast clearing of the calculator inputs. There should be no clutter and all needed info and input should be on the screen with no need to scroll. Portraits should be locked and maintained.

## Define Menu Hierarchy and Traversal The traversal of the menu is simple. When the app is run, the user begins on the Version Selection and Settings Screen where they can select version, adjust settings, or go view the terms and definitions.

## Define Screens and Content 3.5.1 Choose Your Version Selection and Optimization Settings This is the main menu/home screen where the version for the calculator is selected and the settings can be set.

**3.5.2 Ninth Edition Calculator**This screen is where the user can input values used in the 9th edition and get their unit’s average damage as the output. **3.5.3 Eighth Edition Calculator**This screen is where the user can input values used in the 8th edition and get their unit’s average damage as the output.

**3.5.4 View Terms and Definitions**

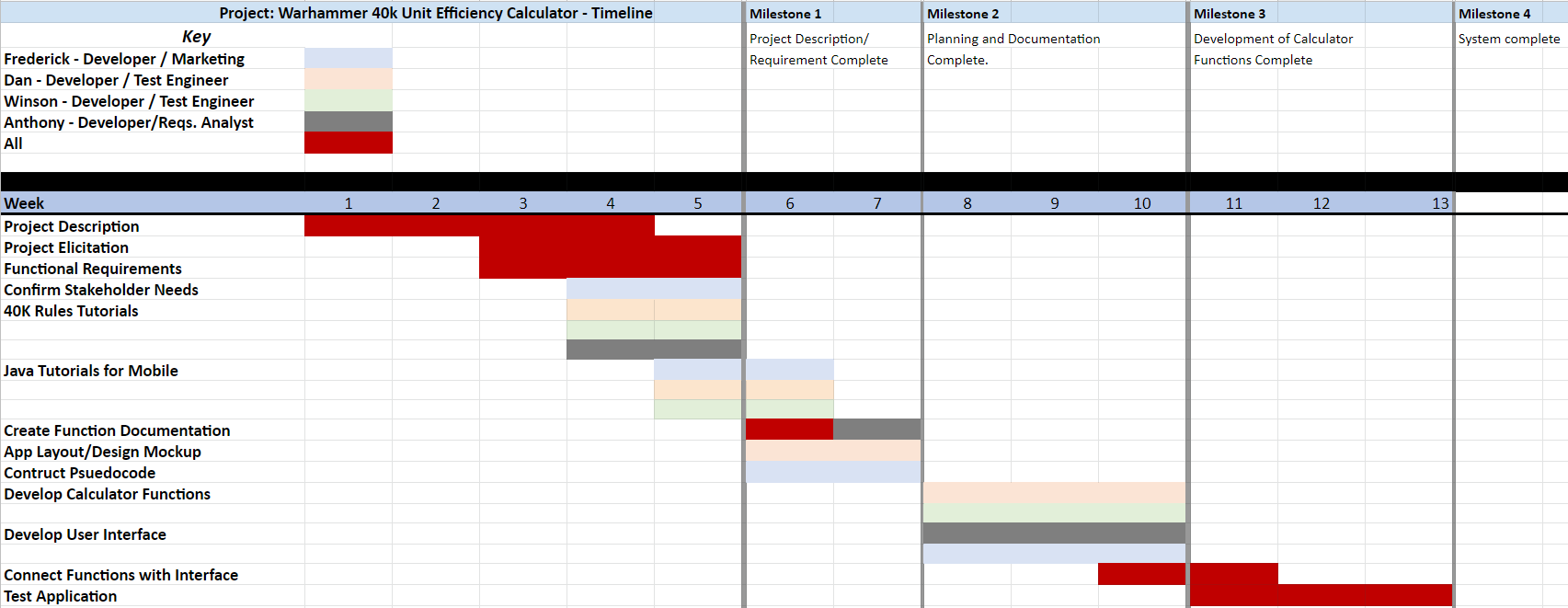
This screen is where the user can view the various terms and 40k statistics that are relevant to the calculator inputs and their definitions.

**3.5.4 Calculator History**This screen is where the user can view the saved calculator input and result history.

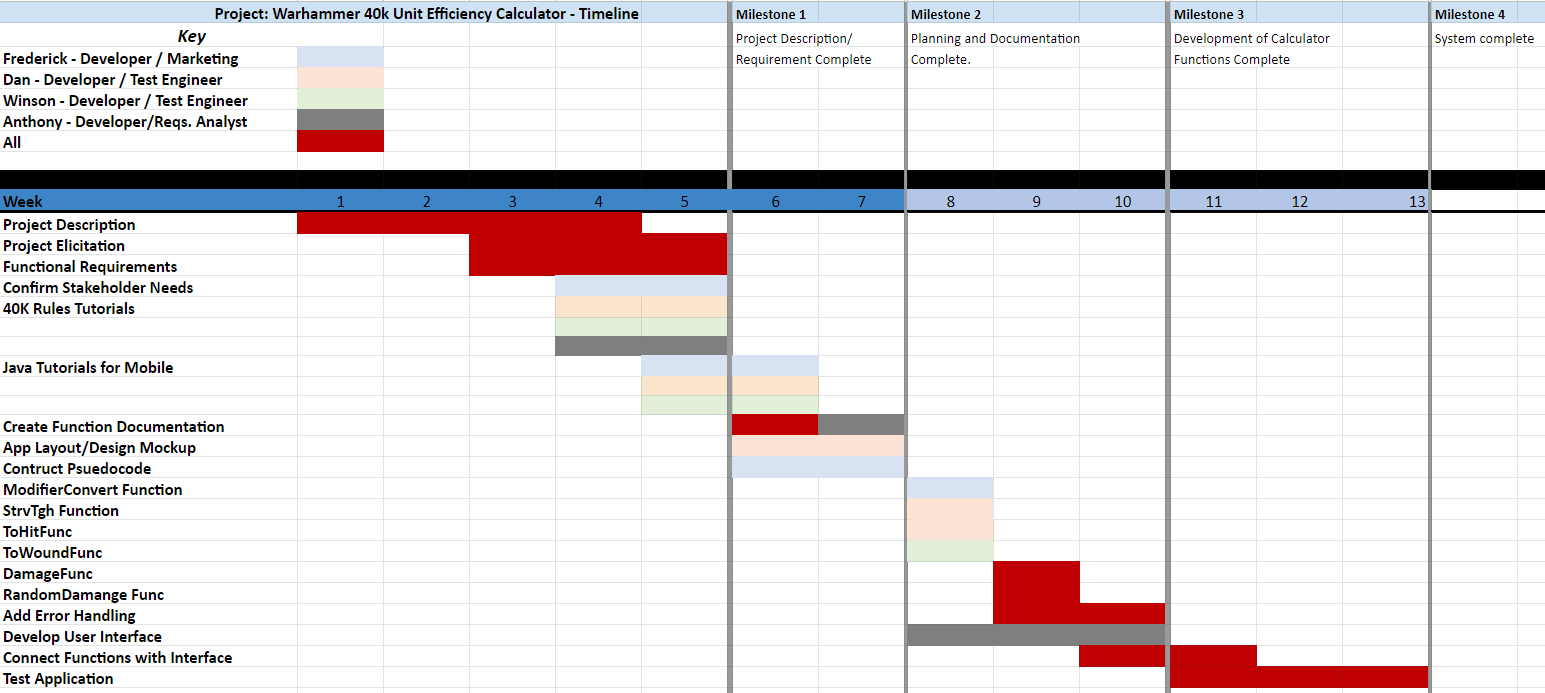
# Development

## Define Reports and Contents

### Progress Report 1 The first progress report introduced what Warhammer 40k was and went over the basic concept and goals of the calculator application. It also went over our requirements, shareholder, technical details, features, updates, and timeline.



### Progress Report 2 The second progress report began with an overview of our flowtop and mockup followed by a tabletop demo. Then we went over the software and communications, progress, limeline, and roadblocks.

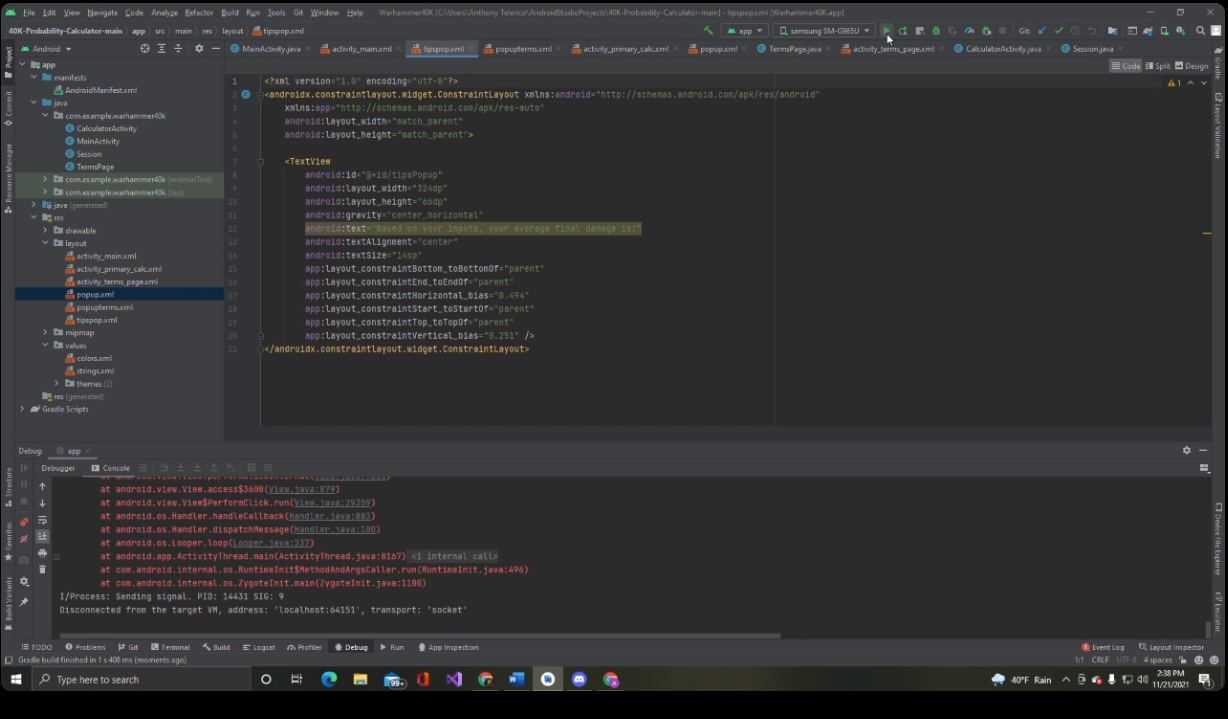
  
  
Progress Report 3  
  
The third progress report discussed our progress with the app development and shows a demonstration of what we had so far in the details of the calculator inputs and steps.

### Progress Report 4 The fourth progress report showcases the UI and working main features of the calculator. The testing and work on the technical doc was also talked about.

## Documentation and Training Guidelines

We are using GitHub to store, push, and pull our latest source code. Our documentation is stored using Google Drive. Our main method of communication via Discord and face-to-face discussion during lab meeting times.

The software used for app development is Android Studio. The application is programmed in Java. Familiarity with these were the skill and training for app development as well as a basic understanding of Warhammer 40k and its rules and mechanics.



# Other Requirements

## Application Functionality

The application’s core purpose and function is to take the user’s inputs and output the average damage value that would be done to an enemy.

## Systems Architecture/Software Design

The application is made to be run on android phones. The testing of the application has been done via Android Studio’s emulator and on Anthony’s android. The software is in Java and designed to be portrait locked and working with android devices specifically most android phones.

## Error Recovery

The majority of errors scenarios that happen are inputs giving incorrect results or actions that crash the app. Various test cases and inputs have been tested to fix errors. Any serious errors were reported over Discord or during meetings and resolved. Errors need to be fixed in code and the emulator needs to be restarted and updated to test. Currently, for error handling in the app, any incorrect inputs will cause a toast message to appear at the button of the screen to notify the user of the error and of the correct range of inputs. All runtime errors are logged in LogCat in the debug terminal which is not accessible by the users.

# Conclusion

## Summary/ Conclusion/Future Works

The Warhammer 40k Unit Efficiency Calculator is a mobile android application coded in Java using Android Studio for the Kent State University’s Software Development Project course. The users are the Warhammer 40k playerbase and the shareholder is Shieldwall Gaming Club. The UI is designed to be simple and fast and the calculator has features for the 9th and 8th Edition and shows the terms and definitions via question button. Additional features include Light and Dark Mode, Terms and Definition, Calculator History, and error handling. Implementation of desired features have been successful within the expected timeline. Roadblocks included familiarizing with Java and Android studio, however, those have been overcome and the result of development has resulted in a satisfactory application that sufficiently meets all shareholder and desired player base needs and requests. Potential Future plans may include the addition of calculators for other other editions as well as further features such as Google Play download, color customization, log entry unit naming, etc.

## Acknowledgements Warhammer 40k is produced by Games Workshop. All members contributed in the programming, presentation, and development, however, Dan did excellent work leading the team and in helping develop and test many of the app features. Frederick advised the team on all Warhammer related matters including testing and requirements. Anthony did a fantastic job on the UI work, app icon and splash screen. Winson wrote the user guide and technical doc. Special thanks to Shieldwall Gaming Club for being our shareholder.