**DEFENCE SERVICES ACADEMY**

**DEPARTMENT OF COMPUTER SCIENCE**

**IMPLEMENTATION OF AN AUTOMATED DYNAMIC WEAPON RANGE VISUALIZATION AND ENGAGAEMENT PLANNING FOR MILITARY TACTICAL SUPPORT SYSTEM USING GEOSPATIAL DATA**

**BY**

**AUNG NYI NYI MIN**

**M.C. Sc Thesis**

**MARCH, 2025**

**PYIN OO LWIN**

ABSTRACT

In modern military operations, precise tactical planning and efficient resource allocation are critical for mission success. This thesis presents the development of an Automated Military Tactical Support System designed to enhance operational planning through dynamic visualization of weapon firing ranges and engagement calculations using geospatial data. Leveraging Geospatial Information Systems (GIS) and advanced algorithmic techniques, the system automates the process of mapping current and target locations, measuring distances, and visualizing weapon ranges based on their power and specifications. The system integrates various APIs to retrieve real-time geolocation data and converts geographic coordinates to the Military Grid Reference System (MGRS) for accurate positioning. Using optimized algorithms, the system calculates the distance from the current location to the desired target, visualizes the effective firing ranges on interactive maps, and identifies the optimal troops and weapons for engagement based on proximity and ammunition requirements. A comprehensive user interface is developed to present these calculations and visualizations intuitively, aiding military personnel in making informed decisions rapidly. Extensive testing with simulated data demonstrates the system's accuracy and reliability, validating its potential to improve tactical planning and operational efficiency in real-world scenarios. This thesis contributes to the field of computer science by integrating GIS and military technology, offering a robust tool for enhanced tactical support. The developed system not only facilitates better resource management but also significantly improves the precision and effectiveness of military engagements.

**Keywords**: Geospatial Information Systems (GIS), Military Tactical Support, Weapon Firing Range Visualization, Distance Calculation, API Integration, Algorithm Design and Optimization, Military Grid Reference System (MGRS), Geolocation Data, Real-time Data Processing, Automated Engagement Planning, Interactive Mapping, Troop and Weapon Allocation, Operational Planning, Resource Management, Tactical Decision Support System

CHAPTER 1

# INTRODUCTION

In the realm of modern warfare, the precision and efficiency of tactical operations are paramount to mission success. The ability to quickly and accurately determine the positioning of forces, calculate engagement ranges, and optimize resource allocation can significantly impact the outcome of military engagements. Traditional methods of planning and executing tactical operations are often time-consuming and prone to human error, necessitating the development of automated systems that can enhance decision-making processes.

This thesis introduces an Automated Military Tactical Support System, designed to leverage advancements in Geospatial Information Systems (GIS) and algorithmic optimization to provide dynamic visualizations of weapon firing ranges and comprehensive engagement planning. By integrating real-time geolocation data through APIs and converting coordinates to the Military Grid Reference System (MGRS), the system aims to deliver precise calculations and intuitive visualizations that support military personnel in making informed decisions swiftly and accurately.

1. Problem Statement

In modern military operations, effective tactical planning and resource allocation are critical for mission success and minimizing collateral damage. However, current methods of tactical planning often rely on manual processes and static data, which are time-consuming, prone to human error, and lack the agility required for real-time decision-making. The challenges in accurately measuring distances, visualizing weapon firing ranges, and dynamically identifying optimal troop and weapon deployment in response to rapidly changing battlefield conditions exacerbate these limitations.

Furthermore, the integration of various data sources, such as geolocation coordinates and weapon specifications, into a cohesive system that provides actionable insights remains a significant challenge. Traditional systems do not efficiently convert geographic coordinates to the Military Grid Reference System (MGRS) or dynamically visualize weapon ranges, limiting the ability of military personnel to make informed decisions quickly. Additionally, the lack of automated tools for calculating necessary ammunition and optimal firing positions further hinders operational efficiency and effectiveness.

This thesis addresses these critical challenges by developing an Automated Military Tactical Support System that integrates real-time geospatial data, advanced algorithms, and intuitive visualizations to enhance tactical planning. The system aims to automate the process of retrieving location data, measuring distances, visualizing weapon ranges, and calculating optimal engagement strategies, thereby providing military personnel with a robust tool for more precise and effective decision-making.

By addressing the limitations of current tactical planning methods, this research seeks to improve the accuracy, efficiency, and responsiveness of military operations, ultimately contributing to mission success and reducing operational risks.

## 1.2 Motivation

The motivation behind this research stems from the need to enhance military decision-making processes with technology that can provide real-time, accurate, and actionable information. Traditional methods of tactical planning often rely on manual calculations and static data, which can be time-consuming and prone to errors. By integrating modern GIS and computational techniques, this system aims to provide military personnel with a robust tool that can significantly improve the speed and accuracy of their operational decisions.

## 1.3 Aim of thesis

The primary aim of this thesis is to develop an automated system that supports military tactical planning by dynamically visualizing weapon firing ranges and optimizing engagement strategies using real-time geospatial data. This system is designed to improve the efficiency and accuracy of military operations, ultimately contributing to mission success.

## 1.4 Objectives of thesis

The objectives of system are as follows;

(a) To integrate APIs for retrieving real-time geolocation data and convert these coordinates to the MGRS system for precise mapping.

(b) To implement algorithms that accurately measure distances between current and target locations using geospatial distance measurement techniques.

(c) To develop a visualization module that dynamically draws weapon firing ranges on maps based on weapon specifications.

(d) To create algorithms that identify optimal troops and weapons for engagement, calculating necessary ammunition and positions.

(e) To design an intuitive user interface that presents all calculations and visualizations to military personnel effectively.

(f) To conduct extensive testing with simulated and real-world data to ensure the system's accuracy, reliability, and practical utility.

# CHAPTER 2

# THEORETICAL BACKGROUND

In this chapter, Flutter, Geospatial Information System (GIS), Military Grid Reference System (MGRS), LatLng to MGRS Conversion, Tactical Decision Support System (TDSS), Bearing Calculation and Mobile Altas Creator (MOBAC) are presented. This thesis is based on GIS and TDSS.

## 2.1 Flutter

Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

## 2.1.1 Key Features of Flutter

1. **Single Codebase:** Flutter allows developers to write code once and deploy it across multiple platforms, including iOS, Android, web, and desktop. This significantly reduces development time and effort.
2. **Hot Reload:** One of Flutter's standout features is hot reload, which enables developers to see the results of their code changes in real-time without restarting the application. This feature accelerates the development process and enhances productivity.
3. **Widget-Based Architecture:** Flutter uses a widget-based architecture, where everything is a widget, including the layout, design, and controls. This approach provides a high degree of customization and makes it easy to create complex UIs.
4. **High Performance:** Flutter's engine is built using C++, and it uses the Skia graphics library, which ensures high performance and smooth animations. It compiles to native ARM code, which further enhances performance on mobile devices.
5. **Expressive and Flexible UI:** Flutter offers a rich set of pre-designed widgets and extensive customization options. This allows developers to create visually appealing and highly responsive user interfaces.
6. **Dart Language:** Flutter applications are written in Dart, a client-optimized programming language developed by Google. Dart is designed for fast development and has features like AOT (Ahead Of Time) and JIT (Just In Time) compilation.

## 2.1.2 Architecture of Flutter

Flutter's architecture consists of three main layers:

1. **Framework Layer:** This is the top layer that developers interact with. It includes the Dart libraries and the widget tree, which provides the building blocks for creating UIs. It is further divided into three sub-layers:

* **Widgets:** This layer contains a rich set of built-in widgets and supports custom widget creation.
* **Rendering:** This layer is responsible for managing the widget layout and rendering them on the screen.

1. **Animation and Gestures:** This layer handles user interactions and animations, making the UI dynamic and responsive.
2. **Engine Layer:** The engine is written in C++ and provides low-level rendering using the Skia graphics library. It also includes the Dart runtime and other essential components for managing animations, gestures, and plugin architecture.
3. **Embedder Layer:** This layer interacts with the underlying platform (iOS, Android, web, or desktop). It provides the platform-specific implementations for rendering surfaces, input events, and other services. The embedder is responsible for launching the Flutter engine and integrating it with the native OS.

## 2.1.3 Use Cases of Flutter

1. **Mobile App Development:** Flutter is extensively used for developing cross-platform mobile applications due to its performance and single codebase feature. Examples include Google Ads, Alibaba, and Reflectly.
2. **Web Development:** Flutter's web support enables developers to build interactive web applications with the same codebase used for mobile apps.
3. **Desktop Applications:** Flutter supports desktop application development for Windows, macOS, and Linux, allowing for a unified development experience across all platforms.
4. **Prototyping:** Due to its rapid development cycle facilitated by hot reload, Flutter is an excellent choice for prototyping and building proof-of-concept applications quickly.

## 2.1.4 Advantages of Using Flutter

1. **Fast Development:** Hot reload, a single codebase, and a rich set of pre-designed widgets accelerate the development process.
2. **Consistent UI Across Platforms:** Flutter ensures a consistent look and feel across different platforms, reducing the effort required for platform-specific adjustments.
3. **Community and Ecosystem:** Flutter has a growing community and a rich ecosystem of packages and plugins that extend its functionality.
4. **Cost-Effective:** By enabling the development of applications for multiple platforms from a single codebase, Flutter reduces development and maintenance costs.

## 2.1.5 Challenges and Considerations

1. **Learning Curve:** Developers new to Dart and the widget-based architecture may face a learning curve.
2. **Large App Size:** Flutter apps tend to have larger binary sizes compared to native apps.
3. **Platform-Specific Features:** While Flutter provides many plugins, there might be a need for writing platform-specific code for certain features not covered by the Flutter framework.

### 2.2 Geospatial Information Systems (GIS)

Geospatial Information Systems (GIS) represent a pivotal technology in the realm of spatial data management and analysis. GIS integrates hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information. This technology has become indispensable across various disciplines, including urban planning, environmental management, logistics, epidemiology, natural resource management, and of course, military operations.

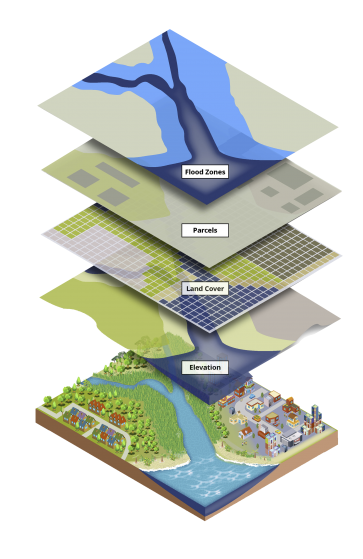


Figure 2.1 Geospatial Information Systems

#### 2.2.1 Components of GIS

GIS comprises several fundamental components:

1. **Hardware:** This includes GPS receivers, satellite imagery sensors, and other data acquisition tools used to collect geographic data.
2. **Software:** GIS software provides the tools for data storage, analysis, and visualization. Examples include ArcGIS, QGIS, and Google Earth Engine.
3. **Data:** Geospatial data includes geographic features (points, lines, polygons), attributes associated with these features, and imagery collected from various sources.

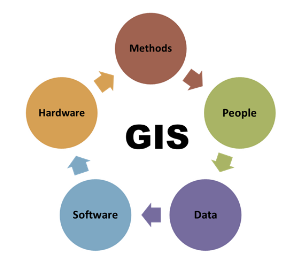


Figure 2.2 Components of GIS

### 2.2.2 Functionality of GIS

GIS performs a wide range of functions essential for spatial analysis:

* 1. **Data Capture:** GIS allows for the capture of geographic data from different sources such as GPS, satellite imagery, and survey data.
  2. **Data Storage:** Geospatial databases store and organize geographic data efficiently for easy access and management.
  3. **Data Manipulation:** GIS enables data manipulation through operations like overlay analysis, buffering, and spatial querying.
  4. **Data Analysis:** Spatial analysis tools in GIS facilitate decision-making by analyzing spatial patterns, relationships, and trends in the data.
  5. **Data Visualization:** GIS produces maps and visual representations that convey complex geographic information in a clear and understandable manner.

### 2.2.3 Applications of GIS

GIS finds applications across diverse sectors:

* 1. **Urban Planning:** GIS aids in land use planning, infrastructure management, and transportation planning by analyzing spatial relationships and demographic data.
  2. **Environmental Management:** GIS supports environmental monitoring, conservation planning, and disaster management by assessing natural resource distribution and environmental impact.
  3. **Public Health:** GIS is used in epidemiology for disease mapping, spatial analysis of health outcomes, and healthcare resource allocation.
  4. **Military and Defense:** In military applications, GIS plays a crucial role in mission planning, intelligence analysis, terrain modeling, and logistics optimization. It enables commanders to visualize terrain features, troop movements, and enemy positions for strategic decision-making.
  5. **Business and Marketing:** GIS assists businesses in site selection, market analysis, and customer segmentation by analyzing geographic data and consumer behavior patterns.

### 2.3 Military Grid Reference System (MGRS)

The Military Grid Reference System (MGRS) is a geographic coordinate system used extensively by military forces worldwide for locating points on the Earth's surface. It is based on the Universal Transverse Mercator (UTM) coordinate system but provides a more concise and standardized method of representing geographic coordinates. MGRS divides the Earth's surface into a grid of zones, each identified by a unique combination of zone number, zone letter, grid square identifier, and numerical location identifier.

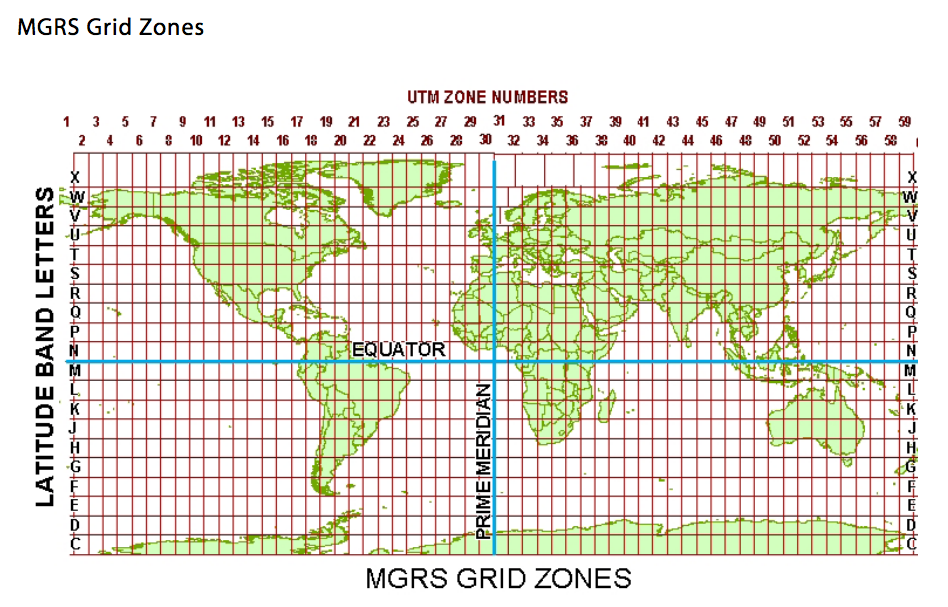


Figure 2.3 MGRS Grid Zones

## 2.3.1 Components of MGRS

* 1. **Zone Number:** MGRS divides the world into 6-degree longitudinal zones, each spanning from 80°S to 84°N latitude. Zone numbers range from 1 to 60, sequentially from west to east around the globe.
  2. **Zone Letter:** Each longitudinal zone is further divided into latitudinal bands using letters "C" to "X" (excluding "I" and "O") for the northern hemisphere and "N" to "X" (excluding "I" and "O") for the southern hemisphere.
  3. **Grid Square Identifier:** Within each 6-degree longitudinal by 8-degree latitudinal zone, the Earth's surface is further subdivided into 100,000-meter squares identified by a two-letter code. These letters represent the southwest corner of the square.
  4. **Numerical Location Identifier:** The numerical location identifier specifies a position within the 100,000-meter grid square using easting (x-coordinate) and northing (y-coordinate) values in meters. These coordinates are derived from the UTM projection system, ensuring accuracy and uniformity.

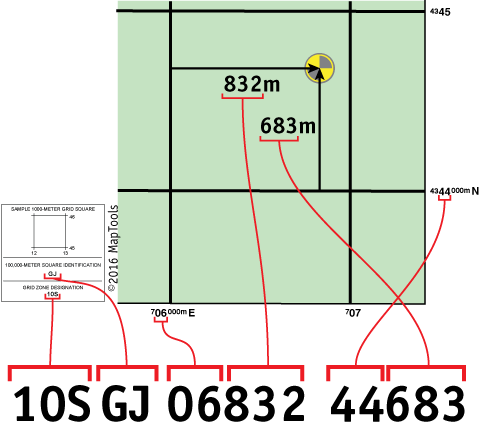


Figure 2.3 Components Of MGRS

### 2.3.2 Equation for LatLng to MGRS Conversion

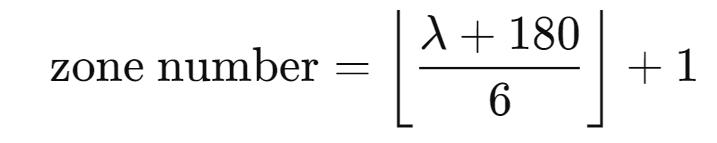
The conversion from LatLng (latitude, longitude) to MGRS involves several steps, including:

* 1. **Convert LatLng to UTM (Universal Transverse Mercator) coordinates:** UTM divides the Earth into 6-degree longitude zones. Each zone has its own coordinate system, which simplifies the conversion process.
  2. **Determine the UTM zone:** Calculate the UTM zone based on the longitude of the LatLng coordinates.
  3. **Convert UTM to MGRS:** Once you have UTM coordinates, convert them to MGRS using the specific MGRS notation rules, which include additional grid squares.

## 2.3.3 Algorithm for LatLng to MGRS Conversion

Here’s a simplified algorithmic approach to convert LatLng coordinates to MGRS:

1. Input**:** Latitude (φ), Longitude (λ)
2. Step 1: Determine UTM Zone
   1. Calculate the UTM zone number based on the longitude (λ):



* 1. Determine the UTM zone letter based on the latitude (φ):
     1. Zones "C" to "M" are for latitudes in the northern hemisphere (φ ≥ 0).
     2. Zones "N" to "X" are for latitudes in the southern hemisphere (φ < 0).

1. Step 2: Convert LatLng to UTM
   1. Convert the latitude (φ) and longitude (λ) to UTM easting (E) and northing (N) using appropriate formulas based on the UTM zone.
2. Step 3: Convert UTM to MGRS
   1. Construct the MGRS string using the UTM coordinates (E, N):

MGRS = zone number +zone letter +grid square identifier + numerical location identifier

* 1. The grid square identifier and numerical location identifier are determined based on the UTM coordinates and specific MGRS rules.

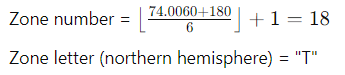
## 2.3.4 Example for LatLng to MGRS Conversion

Let's say we have LatLng coordinates:

Latitude (φ) = 40.7128° N

Longitude (λ) = 74.0060° W

1. Calculate UTM Zone**:**



1. Zone letter (northern hemisphere) = "T"
2. Convert LatLng to UTM**:**

Compute UTM easting (E) and northing (N) using appropriate UTM formulas for zone 18.

1. Convert UTM to MGRS**:**

Construct the MGRS string using zone number, zone letter, grid square identifier, and numerical location identifier.

## 2.4 Tactical Decision Support System (TDSS)

A Tactical Decision Support System (TDSS) is a specialized software application designed to assist military commanders and decision-makers in making informed decisions during tactical operations. It integrates various data sources, analytical tools, and visualization techniques to provide real-time situational awareness and actionable insights. Here, I'll outline the components and algorithmic approach typically employed in a TDSS.

## 2.4.1 Components of a Tactical Decision Support System

1. Data Integration: TDSS integrates data from multiple sources such as:
   1. Geospatial Data: Including maps, terrain models, and satellite imagery.
   2. Operational Data: Mission plans, troop positions, enemy movements.
   3. Environmental Data: Weather conditions, local infrastructure.
2. Data Analysis and Modeling:
   1. Spatial Analysis: Calculates distances, areas, and proximity relationships.
   2. Predictive Modeling: Uses historical data and algorithms to forecast enemy movements or environmental impacts.
   3. Statistical Analysis: Analyzes trends and patterns in operational data.
3. Decision-Making Algorithms:
   1. Route Planning: Determines the optimal paths for troop movements considering terrain and threat locations.
   2. Weapon Allocation: Identifies suitable weapon systems and calculates firing solutions based on target location and weapon capabilities.
   3. Resource Allocation: Allocates resources such as troops, vehicles, and supplies based on mission objectives and operational constraints.
4. Visualization and User Interface:
   1. Map-based Interfaces: Displays tactical maps with overlays of troop positions, enemy locations, and operational boundaries.
   2. Graphical Representation: Uses charts, graphs, and diagrams to present analysis results and decision options.
   3. Real-Time Updates: Supports dynamic updates and real-time data feeds for situational awareness.

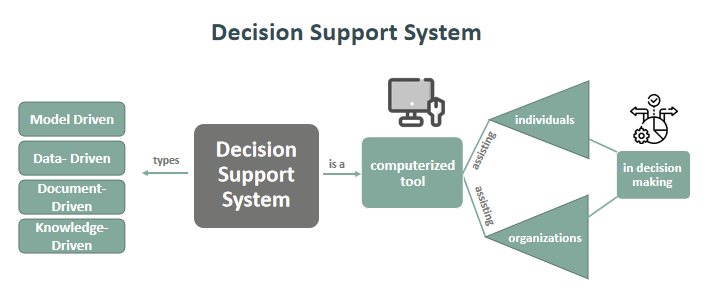
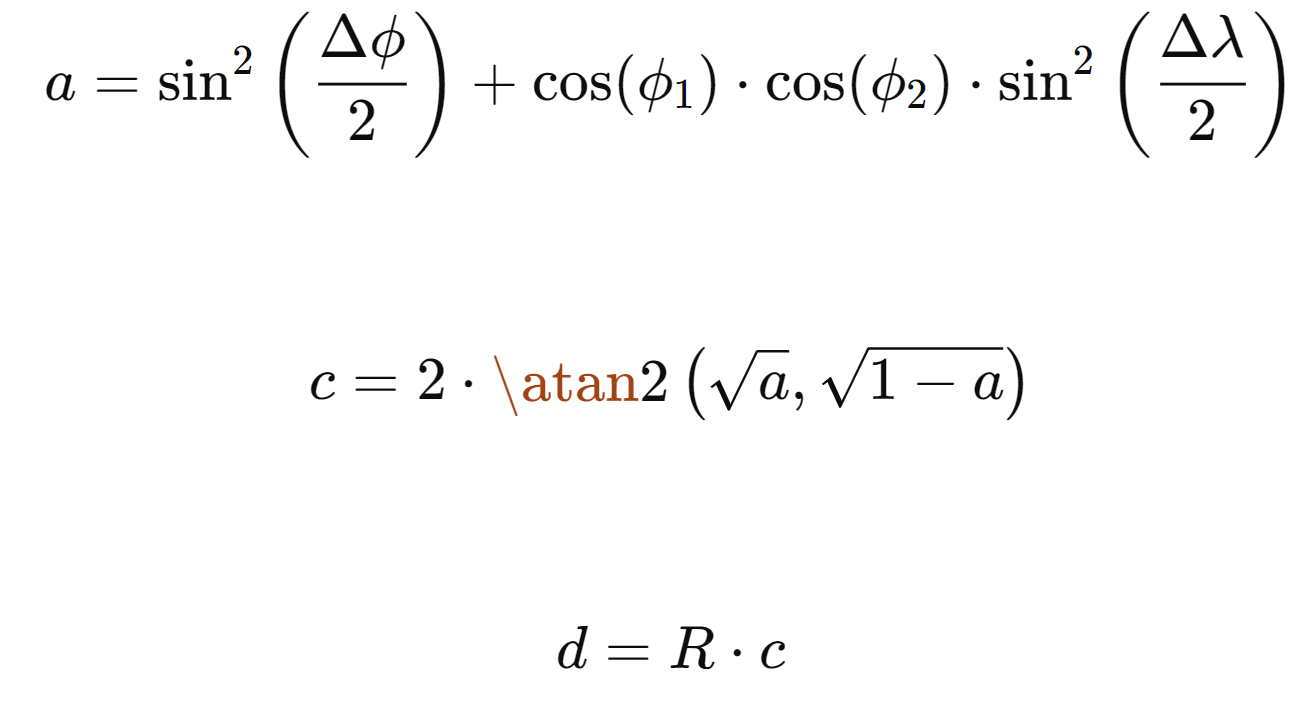


Figure 2.4 Decision Support System

## 2.4.2 General Equations and Algorithms Used in TDSS

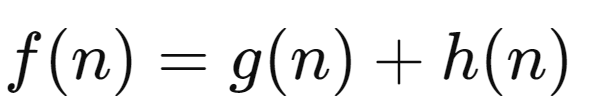
1. Distance Calculation:
   1. Haversine Formula: Used to calculate the great-circle distance between two points given their latitude and longitude coordinates. This is crucial for determining distances between troop positions, target locations, and other critical points on a map.



Where:

* + 1. ϕ1​,ϕ2​: Latitudes of the two points
    2. Δϕ, Δλ: Differences in latitude and longitude between the points
    3. R: Radius of the Earth (mean radius = 6,371 km)

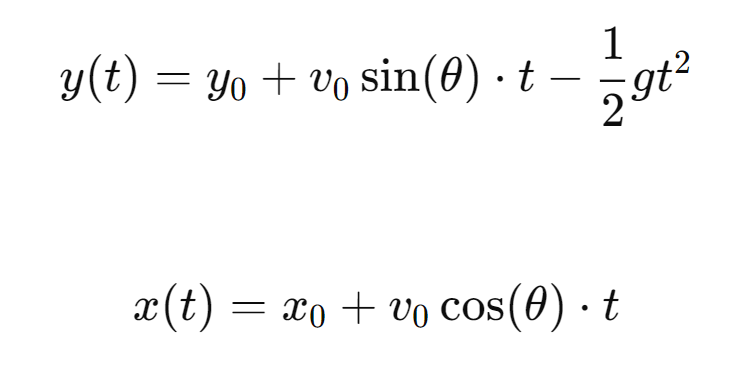
1. Route Planning:
   1. *A Algorithm (A-star):*\* A heuristic search algorithm used for finding the shortest path between nodes (representing locations) on a graph. It considers both the cost to reach a node and an estimated cost to reach the goal (heuristic function), often based on distance or time.



Where:

* + 1. *f(n):* Estimated total cost of path through node *n*
    2. *g(n)*: Cost to reach node *n*
    3. *h(n):* Estimated cost from *n* to the goal based on heuristic

1. Weapon Firing Solutions:
   1. Ballistic Equations: Calculate the trajectory of a projectile fired from a weapon, considering factors such as muzzle velocity, projectile weight, drag coefficients, and atmospheric conditions. These equations typically include components for gravity, air resistance, and initial conditions of the projectile.



Where:

* + 1. (x0​,y0​): Initial coordinates of the projectile
    2. v0​: Initial velocity of the projectile
    3. θ: Launch angle
    4. g: Acceleration due to gravity

1. Spatial Analysis:
   1. Overlay Operations: Used to analyze spatial relationships between different layers or datasets on a map. Operations include intersection, union, difference, and buffering to determine areas of overlap, containment, or proximity between spatial features.

## 2.5 Equation for Bearing Calculation

Given two points:

1. Point 1: (ϕ1,λ1) where ϕ1 ​ is the latitude and λ1 ​ is the longitude..
2. Point 2: (ϕ2,λ2) where ϕ2 is the latitude and λ2​ is the longitude.

The formula to calculate the bearing θ from Point 1 to Point 2 is:

Δλ=λ2​−λ1​

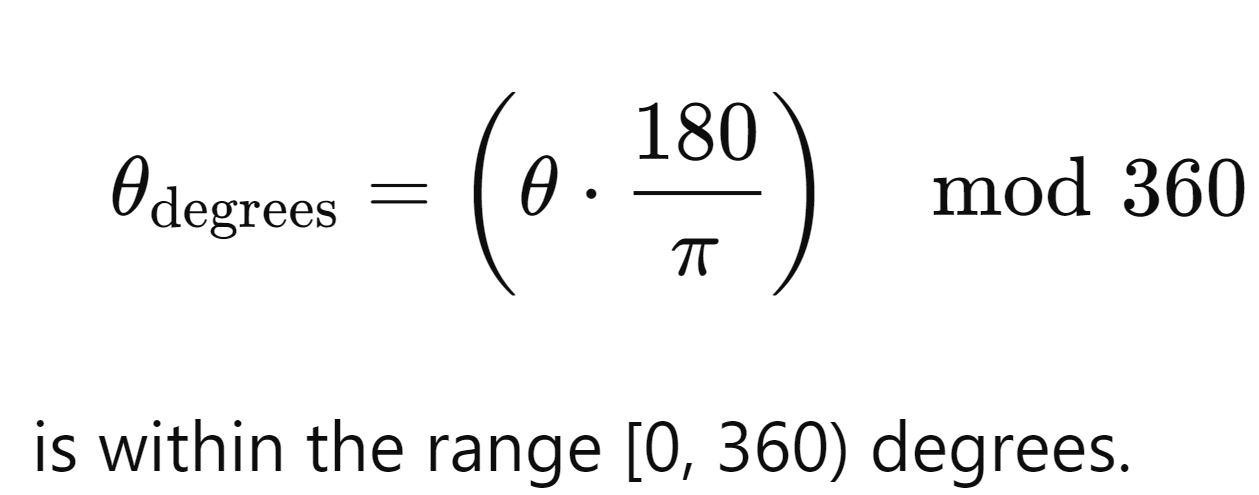
θ=\atan2(sin(Δλ)⋅cos(ϕ2​),cos(ϕ1​)⋅sin(ϕ2​)−sin(ϕ1​)⋅cos(ϕ2​)⋅cos(Δλ))

Where:

1. ϕ1 ​ and ϕ2 ​ are the latitudes of the two points in radians.
2. λ1 ​ and λ2 are the longitudes of the two points in radians.
3. Δλ is the difference in longitude.
4. \atan2(y,x) is the two-argument arctangent function, which returns the angle in radians between the positive x-axis and the point (x, y).

## 2.5.1 Conversion to Degrees

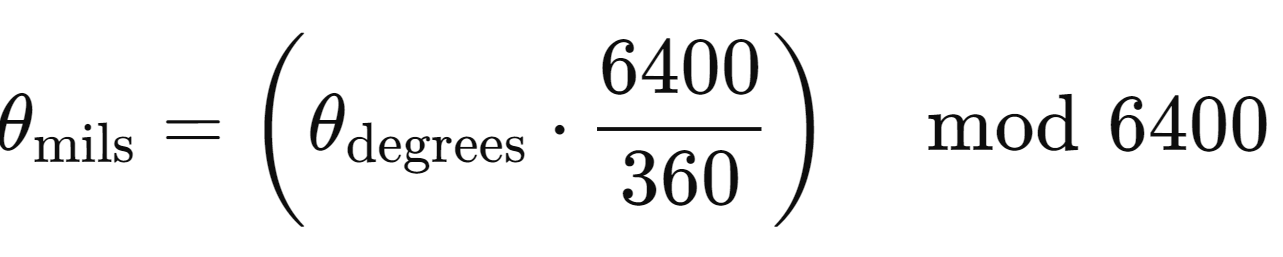
The bearing θ is initially calculated in radians. To convert it to degrees, use the following formula:



This ensures that the bearing is within the range [0, 6400) mils.

## 2.5.2 Conversion to Mils

To convert the bearing from degrees to mils (NATO mils, where 1 mil = 1/6400 of a circle):



This ensures that the bearing is within the range [0, 6400) mils.

## 2.6 Mobile Atlas Creator (MOBAC)

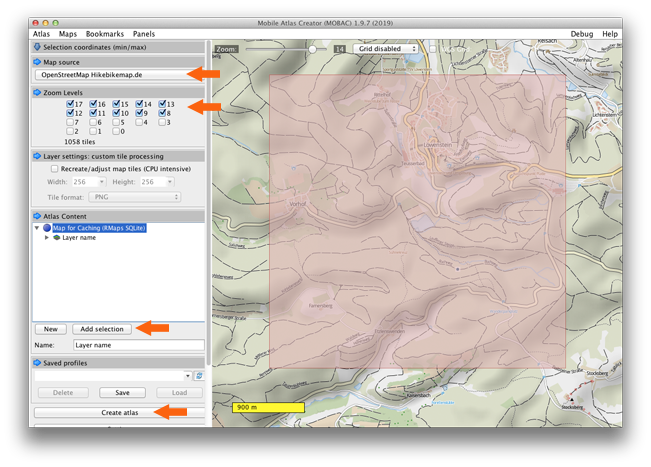
Mobile Atlas Creator (MOBAC) is a free and open-source application used to create offline maps for mobile applications. It allows users to download map tiles from various online sources and compile them into atlases, which can then be used offline on devices such as smartphones, tablets, and GPS units. This is especially useful for activities where internet access is unreliable or unavailable, such as hiking, cycling, or traveling in remote areas.

## Key Features of MOBAC

1. **Multiple Map Sources:** Supports a variety of map sources, including OpenStreetMap, Google Maps, Bing Maps, and custom map servers. This allows users to choose the best map for their needs.
2. **Customizable Atlas Formats:** Generates atlases in multiple formats compatible with different navigation applications and devices, such as OruxMaps, Locus Maps, RMaps, TrekBuddy, and more.
3. **Offline Maps:** Creates offline maps that can be used without an internet connection, providing users with reliable navigation tools in areas with poor or no connectivity.
4. **Flexible Map Selection:** Users can define the area to be included in the atlas by drawing a rectangle on the map, entering coordinates, or importing GPX/KML files to specify a route or area.
5. **Zoom Levels:** Allows selection of multiple zoom levels to ensure detailed mapping at various scales, from a broad overview to street-level details.
6. **Tile Management:** Efficiently manages map tiles by downloading and storing only the necessary tiles, optimizing storage usage on the device.
7. **Cross-Platform:** Available for Windows, macOS, and Linux, making it accessible to a wide range of users.

## 2.6.2 Applications of MOBAC

1. **Outdoor Activities:** Ideal for hikers, cyclists, and adventurers who need reliable maps in remote areas without internet access.
2. **Travel:** Useful for travelers who want to avoid roaming charges and ensure they have maps available in regions with limited connectivity.
3. **Emergency Services:** Can be used by rescue teams and emergency services to navigate areas affected by natural disasters where infrastructure is compromised.
4. **Military Use:** Supports military operations by providing offline maps for planning and executing missions in remote or hostile environments.



## 2.7 Summary

The development of an Automated Military Tactical Support System (AMTSS) leverages geospatial data to enhance the capabilities of military operations by providing dynamic weapon range visualization and engagement planning. This system integrates real-time geospatial data with advanced algorithms to facilitate precise decision-making, optimize resource allocation, and ensure effective engagement strategies in complex and rapidly changing operational environments.

The development of the Automated Military Tactical Support System represents a significant advancement in military technology, integrating geospatial data and advanced computational techniques to support dynamic weapon range visualization and engagement planning. By enhancing situational awareness, optimizing resource allocation, and supporting real-time decision-making, the AMTSS plays a critical role in modern military operations, ensuring that commanders have the tools and information they need to achieve strategic objectives and maintain operational superiority in complex and dynamic environments.

# CHAPTER 3

# SYSTEM DESIGN AND IMPLEMENTATION

This chapter presents the system design, detail design, database design and implementations of the system.

## 3.1 Work Flow of the System

There are three roles in the work flow of the system. They are Services repository, visitor profiles and planning a personalized trip. This system considers each player's skill level, making it fair for everyone with the rules of handicap and System 36 playing based on a real-time web-based golf tournament.

Weapon Range Data (Weapon Specs, Range Info)

Communication & Coordination Network

Geospatial Data Acquisition & Module

TDSS Interface (Command Center)

Engagement Planning Module (Scenario Simulation)

Dynamic Visualization Engine

(Real-Time Maps, Range Rings)

Feedback & Adjustment Loop (Real-Time Data Processing)

Figure 3.1 Work Flow of the system

## 3.2 Overall Design of the System

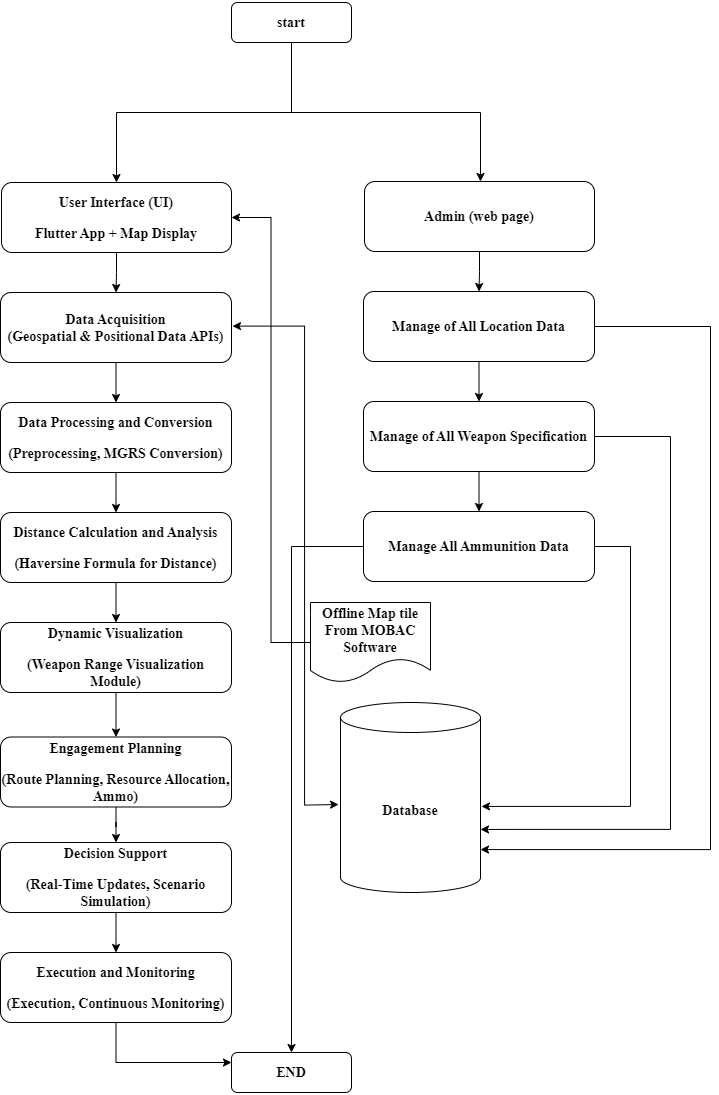


Figure 3.2 Overall System Design

Figure 3.2 shows the overall processes of system. This system integrates various data sources and processing modules to provide a comprehensive tool for military tactical support. It enhances situational awareness, optimizes engagement planning, and supports real-time decision-making and execution monitoring, ensuring that military operations are conducted efficiently and effectively.

## 3.3 Detail Design for Admin

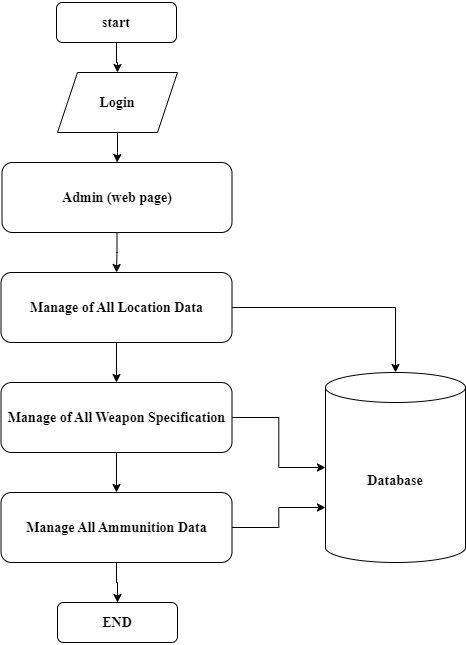


Figure 3.3 Detail Design for admin

## 3.4 Detail Design for Flutter UI

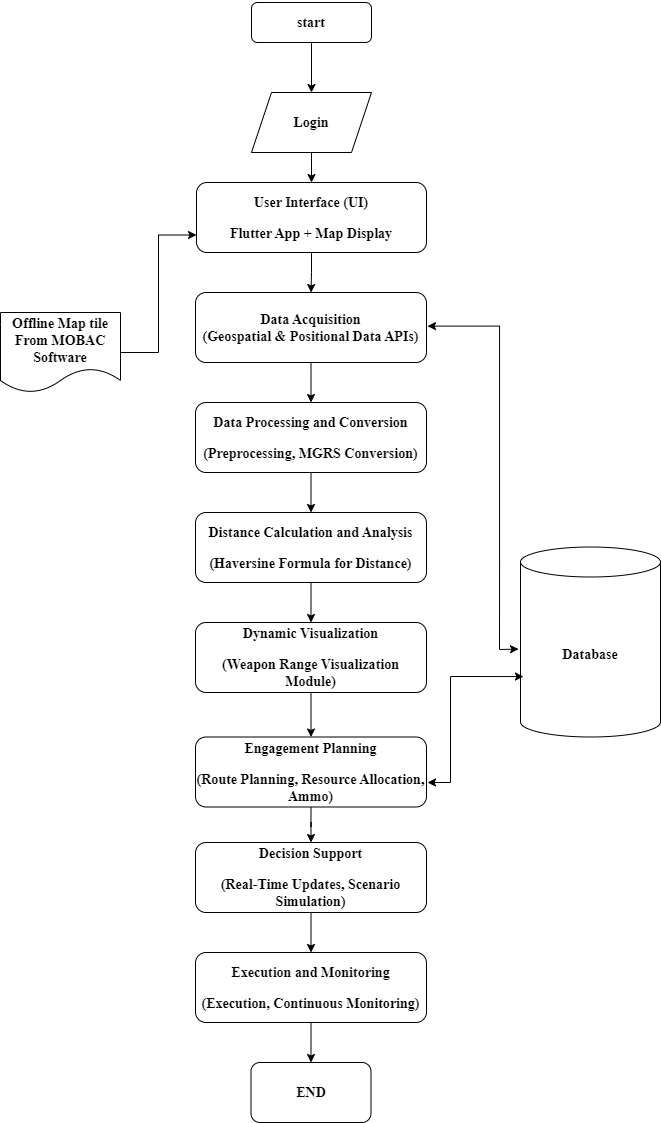


Figure 3.4 Detail Design for Flutter UI