ClimbPlanner

Requirements Analysis Report

By

Quattro Group

02.12.2008

TABLE OF CONTENTS:

1. Introduction……..2

1.1 Project Definition…….....2

1.2 Project Scope and Goals…….....2

2. Process…….....3

2.1 Team Structure…….....3

2.2 Process Model…….....3

3. Project Schedule/Gantt Chart……4

4. Project Requirements…….....5

4.1. System Requirements …….....5

4.1.1. Hardware Requirements…….....5

4.1.2. Software Requirements…….....6

4.2 User Requirements…….....7

4.2.1 Use Case Diagram…….....7

4.2.2 User Functionality…….....8

4.2.3 Nonfunctional Requirements …….....9

4.2.3.1. Usability…….....9

4.2.3.2. Portability…….....9

4.2.3.3. Reliability…….....10

4.2.3.4. Security…….....10

5. System Features……..10

5.1. User Interaction Engine……..10

5.2. Web Engine……..11

5.3. Planning Engine……...11

5.4. Simulation Engine……12

5.5. Report Engine……...12

5.6. Intercommunication Engine……...13

6. Scenario…….....13

7. Conclusion…….....14

8. Diagrams……..14

8.1. Entity Relationship Diagram……..14

8.2. Dataflow Diagrams……..15

1. Introduction

1.1. Project Definition:

A geographic information system (GIS) integrates hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information. In real life, GIS has a lot of usage areas like:

- Navigation Systems

- Military Applications

- City and Region Planning

- Logistics

- Agriculture

- Real Estate

- Mining

- Oceanography etc.

We will try to add a new point of view by using GIS in climbing. With this application, we are planning to assist climbers before and while the climb. Climbing is mainly done by groups and members of group affect route and time. Moreover, the equipment of group has a significant effect on climbing. Mainly, amateur groups try routes which are recommended by professionals. However nobody can assure that the group is capable of climbing the recommended route. Thus our aim is to produce a program that helps climbers to define their safe path to peak.

We are planning to use power of GIS to show users 3D simulation of the climbing area. This 3D simulation will be based on a predefined height map or the height map provided by users. The program will define the route by considering group members, weather conditions and terrain conditions. As a result, users will be able to see what they will encounter during climb, before the climbing activity by wandering on the produced 3D image.

1.2 Project Scope and Goals:

In this project we aim to followings:

- to assist climbers during defining their route,
- to provide options for route considering safest or shortest one,
- to visualize the result with vector maps,
- to rotate/zoom in/zoom out/scroll the returned map,
- to make some estimations about climbing duration
- to give information about camping places
- to provide the user with ease of use

In this project, we will focus on these topics:

- to organize a complete project,
- to produce a program in a user-friendly manner,
- to implement effective shortest path algorithms,
- to manipulate map data in GIS part,
- to have a detailed 3D visualization.

2. PROCESS

2.1. Team Structure:

Since we have been working together for a long time, we decided that Democratic Decentralized (DD) fits best to our project. Each member in the group has a talent in different areas. Therefore, each member can make some suggestions about their research and their area of specialization. At the end, the suggestion will be voted and decision will be taken.

2.2. Process Model:

Our project consists of the following predefined steps:

- Requirements Analysis
- Initial Design

- Detailed Design

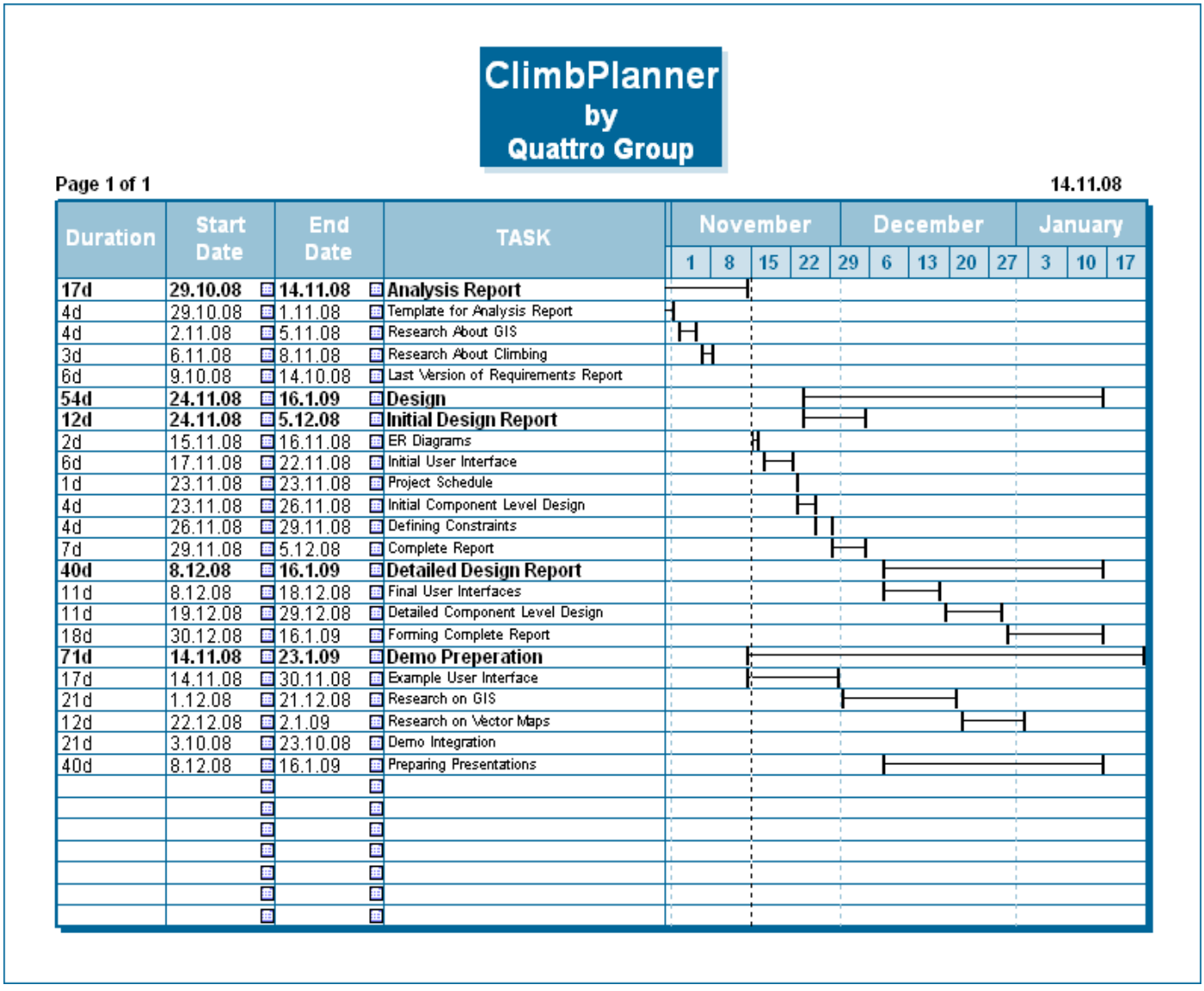
- Testing – Debugging

While deadlines of these steps are determined, we will release several prototypes in implementation stage and we have to consider their feedbacks. Thus, we may return back to some parts of detailed design according to the success of the prototypes. As a result, we decided that iterative process model best suits to our project.

3. PROJECT SCHEDULE

3.1 Gantt Chart:

Following Gantt Chart shows only plan of one semester time:



4. PROJECT REQUIREMENTS

4.1. System Requirements:

4.1.1. Hardware Requirements

Hardware requirements for our project comprise two main groups; the former of which is developer side requirements and the latter one is client side requirements

Developer-Side Hardware Requirements

Approximate minimal requirements for the development of our project is

- 1.8 GHz CPU
- 512 MB DDR2 RAM
- 60 GB HDD
- 128 MB Graphics Card
- Internet Connection

Client-Side Hardware Requirements

Since the clients will not make complex operations, the processor does not need to be very fast. However, it will be beneficial that it has a good graphics card and enough processing power because the clients will see the simulation of the terrain and the route on their screen. The clients don't need to have internet connection because predefined maps and images for the route will be supplied to them inside the program, but if they have internet connection, it may be useful for downloading new maps.

Hence the approximate minimal requirements for the client side is

- 1.6 GHz CPU
- 512 MB DDR2 RAM
- 40 GB HDD
- 128 MB Graphics Card

4.1.2. Software Requirements

During the analysis, design, implementation and testing phases of the project, we will use several tools to carry out the project. These tools can be divided into two main groups; first one is documentation tools and the second is development tools.

Documentation Tools

We will use several documentation tools, particularly throughout the first semester. We have two opportunities for word processing; either Microsoft Office Word 2003 or Microsoft Office Word 2007 combined with Adobe Acrobat Professional, or OpenOffice 2.0 Writer.

We have chosen SmartDraw as the drawing tool because of its ease of use. We have some experiences from the previous terms also. In this phase of the project, we have drawn data flow; use case, entity relationship diagrams using SmartDraw. For drawing Gantt Chart, we preferred using Milestones Professional 2008.

Development Tools

After a long research period, we decided to use Java 2 Platform Standard Edition (J2SE). In this critical choice, there are several reasons. First of all, platform independency is an important factor for this choice. Java technologies are platform independent and they are very compatible with almost every Linux distribution, Windows, MacOS and Solaris. Secondly, we have some experience in J2SE, from previous courses we have taken. Thirdly, there are very successful Integrated Development Environments (IDEs) for Java. Eclipse is our favorite. We may also use NetBeans, since it is a very nice and complete solution for Java development. Since compatibility is provided by Java, we can use both of the IDEs at the same time. As a final reason, J2SE is too widespread and popular that we are sure that we can find help on the internet easily.

For DBMS usage, most important factor is the platform independency. So, we have strong options such as MySQL and PostgreSQL. We will use MySQL most probably because it is easy to use, very widespread and also because we are experienced on it.

We will use GeoTools as Java GIS toolkit. It is an open source (LGPL) Java code library which provides standards compliant methods for the manipulation of geospatial data. We will use GridCoverageExchange interface to access grid coverage (raster) data formats in GeoTools. The first reason why we have chosen GeoTools is that, it is a widespread OpenSource Java toolkit. The second reason is that, it supports lots of data formats. For example;

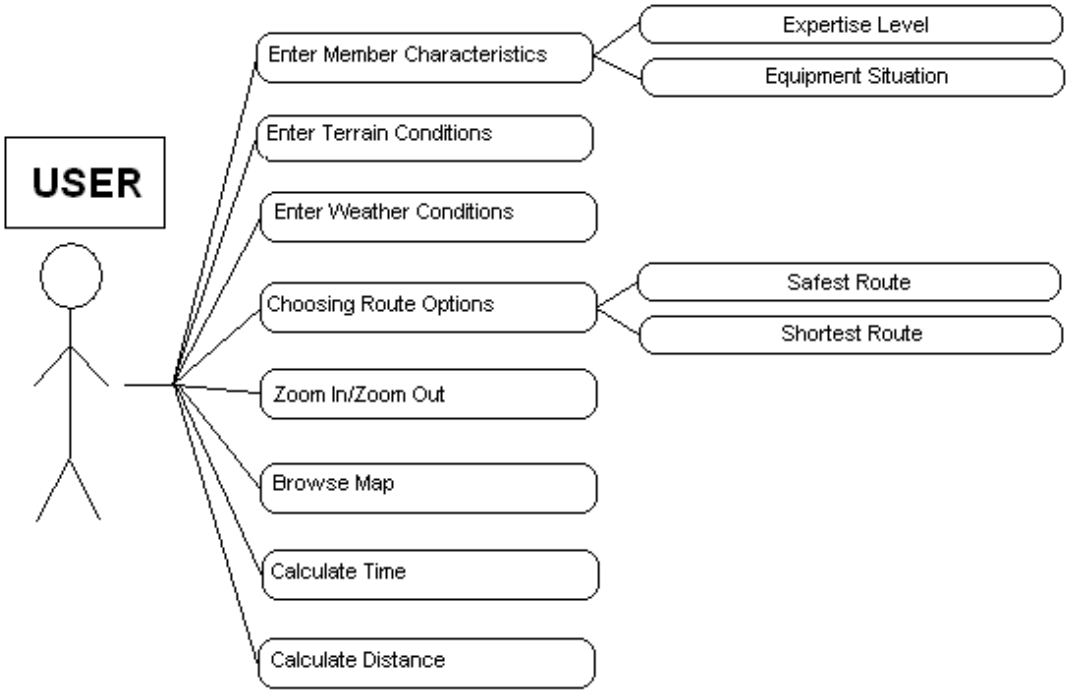
- Shapefile - an ESRI shapefile (R/W)
- GML - Geography Markup Language (R)

- WFS - Features from an OGC Web Feature Server (RW)
- PostGIS geometric objects for PostgreSQL(R)
- Oracle Spatial - Oracle's extension for spatial data (R)
- ArcSDE - ESRI's middleware for spatial databases (R)
- MySQL - support for the new geometry types
- GeoMedia - an Intergraph format (R)
- Tiger - Topologically Integrated Geographic Encoding and Referencing developed at the US Census Bureau (R)
- VPF - Vector Product Format, a data interchange format (R)
- MapInfo- MIF (Mapinfo Interchange Format) (RW)

In addition, we may benefit from using APIs of Google Earth and NASA World Wind for simulating the map chosen by the developer. We can serve information obtained from these programs, about the area chosen by the user. However, we haven't decided on whether we will use this information, for now.

4.2 User Requirements:

4.2.1 Use Case Diagram



Use case diagram for user

4.2.2 User Functionality

Our program is planned to run on users' computer. The program will need internet connection if user specifies the map with a name or coordinate. Mainly the working principle of the program is after creating of a group of mountaineers and choosing an area to climb, deciding on the route according to group's restrictions and terrain restrictions.

- **Adding New User**

Users can add themselves to the database. This database holds data about users. The user either inputs data about himself/herself or inputs data about other group members. The attributes of the user will be user id, name, age, gender, experience, physical condition, equipment condition, maximum climbing speed, rest time (specifying the time that a user should rest before continuing the climbing).

- **Adding New Map**

Users will add new maps to database. These maps will be hold in database for future use. Although, there are a lot of map file types on the internet, we choose DTED data type because of its very wide usage area. The program will process the data provided and 3D visualization of map will be seen.

- **Member Characteristics**

User will define members of group in means of age, health, expertise level, gender. Moreover, equipment of group will be provided by user in this part. Considering these, program will calculate best (shortest/safest) route for group. Program will provide options to select which route user want to use.

- **Minimum Requirements for Route**

In this part of project, program shows user different routes. User will choose the route and the output of the program will be minimum requirements for this route like equipment, expertise level, number of members and information about which seasons the route is available for climbing.

- **Terrain Conditions**

Because the height characteristics of climbing area are defined by map data, only condition of the surface like ice, rock, mud and slippery soil will be provided by user.

- **Weather Conditions**

Season has a big effect over route to be chosen. Some routes are available for climbing in only some seasons. Program will determine route considering the weather conditions.

- **Display Items on 3D Map Screen**

That is main user functionality of the program. Program will determine routes considering all information provided by user. User will be able to see routes and choose between them. All potential routes will have a safety degree. This safety degree will be determined by height characteristics of routes, equipment conditions and characteristics of members of group. Moreover, on the route there will be camping places again determined by program. Determination of these camping places and time interval between them will be done by taking consideration of the whole situation.

User can rotate the display, zoom in/zoom out on the display and see 3D digital visualization of map. This visualization will be as real as in the limits of map provided by user.

4.2.3 Nonfunctional Requirements

4.2.3.1. Usability:

The usability of our product is very important for us. We are planning to develop a product to be user friendly for all of users without any distinctions depending on the experience and knowledge of the users. Our product will lead climbers in their route to their safe and enjoyable trip to the peak of the mountain. The interfaces of our product will be very clear and understandable. Since our product will be system independent it will be able to run on Windows, Linux, MacOS, Solaris etc. Hence we do not think the users will encounter any problem while using it.

4.2.3.2. Portability:

We can definitely say that our product will be portable, because during the development process we will always use platform independent tools. For instance, J2SE, MySQL, GeoTools are all open source platform independent technologies. Thus, our program will work on any operating systems that support these technologies.

4.2.3.3. Reliability:

We are planning to develop our product so stable that any minor problem will not cease the program. Moreover, we are going to do all of the possible tests after implementation to minimize the bugs on the program. We are planning to use programs such as Bugzilla in order to report and trace the bugs effectively.

4.2.3.4. Security:

There is no distinction between users in ClimbPlanner program such as administrator, moderator or user. Hence, there is no need for making password or fingerprint controls and as a result there will not be any hierarchy difference between users. Security will be supplied by the user of the program and no responsibility will be taken for unauthorized access if any.

5. System Features

System features illustrate organizing the functional requirements for the product and the major services provided by the product.

5.1 User Interaction Engine

5.1.1 Description

There are several types of climbers in the real world with different levels of experience. To be able to identify each climber and find out their level of experience in the system, information about the users will be stored in the database. Moreover, in this module, the moderator will enter the group information. The following part describes the requirements about user management.

3.1.2 Functional Requirements

UIE-1: This sub-system will store ID numbers of mountaineers.

UIE-2: This sub-system will store climbing speed of mountaineers in meters per hour.

UIE-3: This sub-system will store rest time of mountaineers in minutes.

UIE-4: This sub-system will store physical conditions of the mountaineers differentiating between values 1–10.

UIE-5: This sub-system will store experiences of mountaineers differentiating between values 1–10.

UIE-6: This sub-system will create a climbing group with a group id.

UIE-7: This sub-system will allow the moderator to select mountaineers for a group.

5.2 Web Engine

5.2.1 Description

Communication with the Web Mapping Server and Web Featuring Server will take place in this engine. The main capabilities of this module are described in the following part.

5.3.2 Functional Requirements

WE-1: This sub-system will request the data of climbing area defined by user.

WE-2: This sub-system will provide DTED and GEOTIFF files from the web server.

WE-3: This sub-system will provide vector data from the web server.

WE-4: This subsystem will provide raster data from the web server.

WE-5: This sub-system will provide weather conditions from the web server.

5.3 Planning Engine

5.3.1 Description

Planning engine will be the main part of system because all the determination about the route and needed info is realized here. The following part will describe the requirements about planning engine.

5.3.2 Functional Requirements

PE-1: This sub-system will compute the safest route of the climbing activity.

PE-2: This sub-system will compute the shortest route of the climbing activity.

PE-3: This sub-system will calculate the time of the climbing activity.

PE-4: This sub-system will determine the camping locations of the climbing activity.

PE-5: This sub-system will determine the required equipments for the climbing activity.

5.4 Simulation Engine

5.4.1 Description

Simulation of the route (according to given inputs), camping areas and further interaction will be realized in this engine. The following part will describe how the system displays visual outputs and enables users to interact with those outputs.

5.4.2 Functional Requirements

SE-1: This sub-system will be capable of displaying the route on the map.

SE-2: This sub-system will be capable of displaying camping areas by highlighting the area and indicating with text.

SE-3: This sub-system will have zooming capability.

SE-4: This sub-system will allow users to change the direction and angle of the camera in order to show the area with detail.

SE-5: This sub-system will be capable of adjusting the speed of the simulation.

SR-6: This sub-system will be capable of pausing the simulation and taking screenshots.

SR-7: This sub-system will be capable of replaying the simulation.

5.5 Report Engine

5.5.1 Description

In this module of the system, textual information will be provided to the users. The following part describes what composes information that the system will report.

5.5.2 Functional Requirements

RE-1: This sub-system will be capable of reporting safest route of the climbing.

RE-2: This sub-system will be capable of reporting shortest route of the climbing.

RE-3: This sub-system will be capable of reporting the required equipments for the climbing activity.

RE-4: This sub-system will be capable of reporting camping locations.

RE-5: This sub-system will be capable of reporting the timing of the climbing activity (i.e. beginning time, finishing time and resting time).

RE-6: This sub-system will allow users to specify the format of the report.

5.6 Intercommunication Engine

5.6.1 Description

This module will be an interface among the other engines of the system that will provide communication and inter-operations. The following part describes the capabilities of this module.

5.6.2 Functional Requirements

ICE-1: This sub-system will be capable of evaluating group information such as experience and skill levels of the group.

ICE-2: This subsystem will combine report and 3D simulated data.

ICE-3: This subsystem will decide on connecting the web server or gathering the map from the local database.

ICE-4: This sub-system will gather vector-raster data of map from determined location.

ICE-5: This sub-system will evaluate 2D projection of the route for report.

ICE-6: This sub-system will gather route, time and map data.

6. Scenario

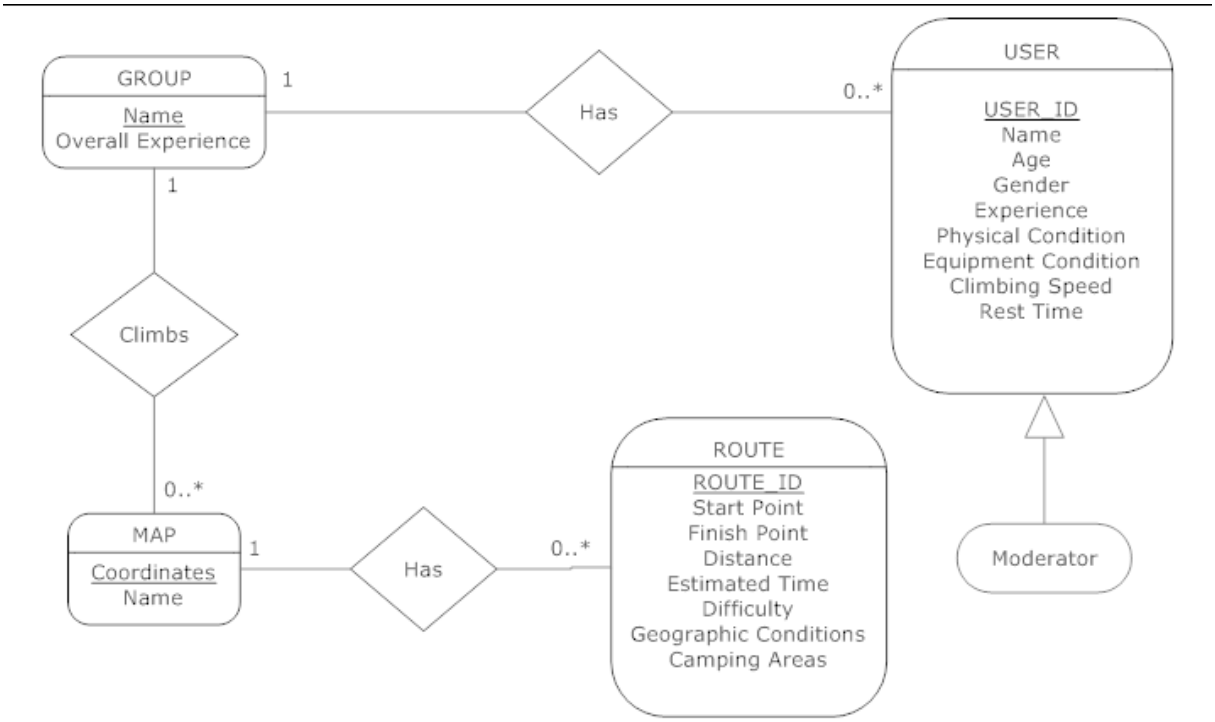
We tried to form a scenario where this program is used in a very important situation. One day an amateur climbing group decides to climb to a mount that they haven't climbed before. They use our program and the program gives them 2 route options. 1 of them is the safest one, the other is shortest one. They choose the shortest one. They start climbing and because the season is winter it becomes harder and harder to move on. After they reach the peak, on the return way one of the members of group slides and stuck between rocks. They need to do something before the member freeze to death. Other members inform authorities about the accident, where it happened, and situation of the injured member. Rescue team prepares to climb to the area but before the climbing they need to know how they can reach there in the shortest time and how they can return in the safest way. After providing information about situation, the program will define their path to accident area and path for return back. Using the path, rescue team arrives in the accident area on time and thanks to the ClimbPlanner program the climber survives.

7. Conclusion

During the preparation of the report we noticed some details about project. These details need to be defined in a detailed way before we reach at a conclusion. Meanwhile, we defined necessities of program and users to determine functionalities of program. Before final prototype demo we will try to make some prototypes and test the program against the requirements defined above.

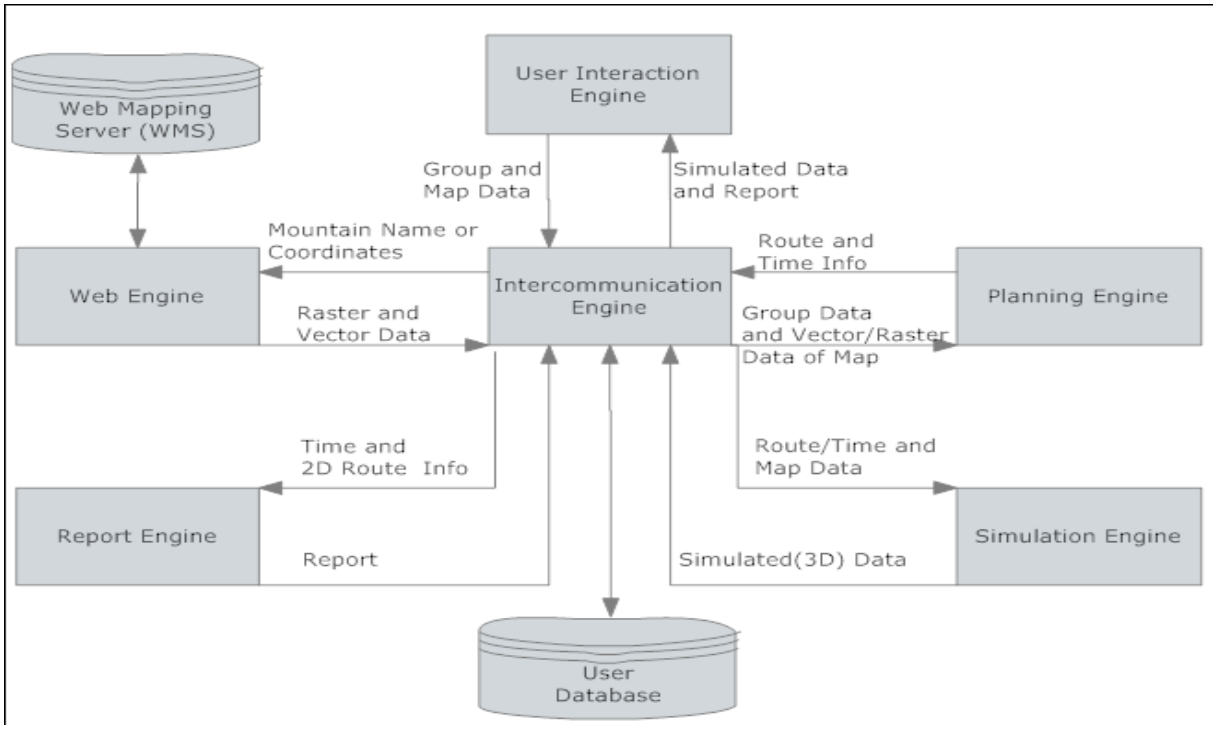
8. Diagrams

8.1 Entity Relationship Diagram



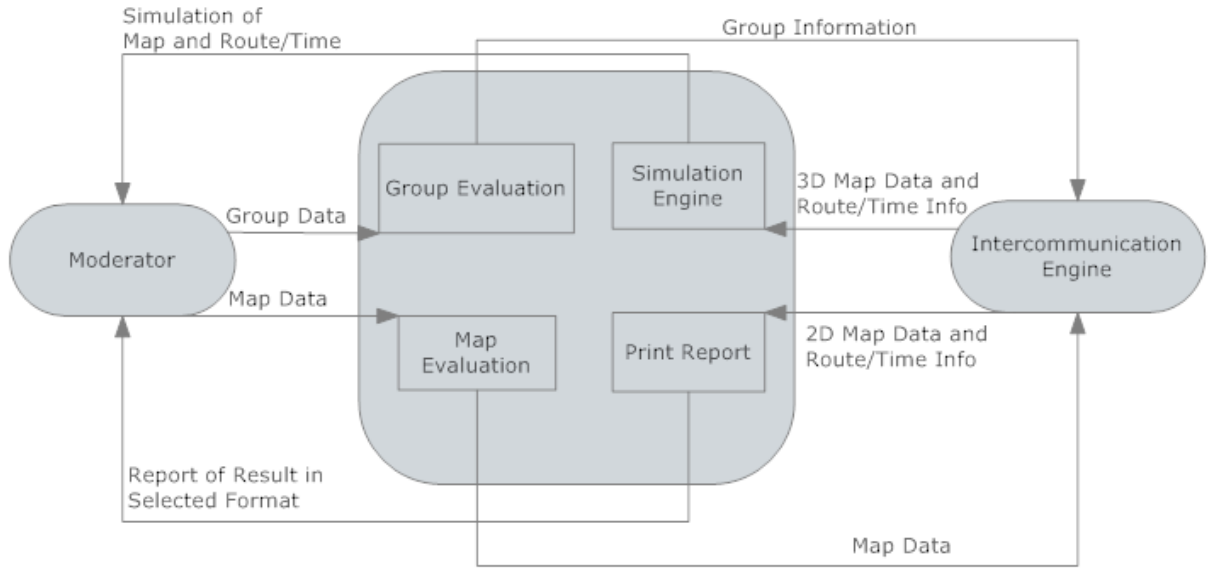
8.2 Data Flow Diagrams

Level 0

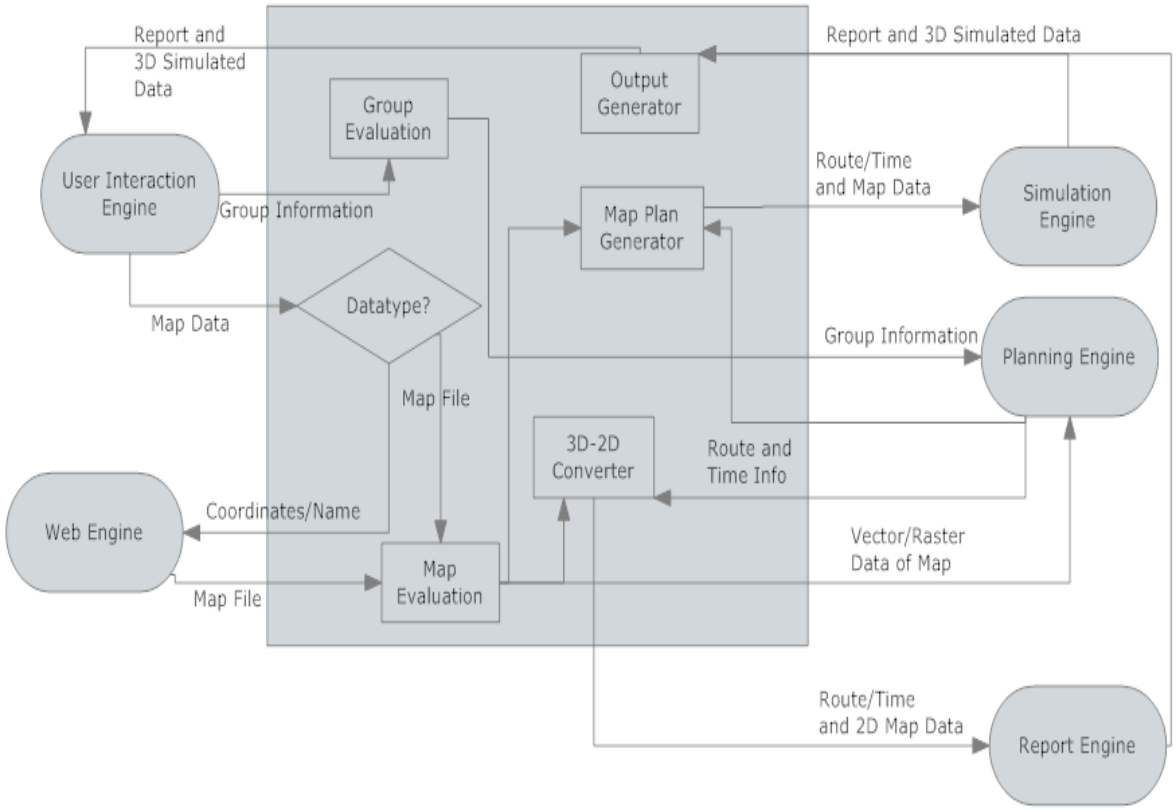


General System Capabilities

Level 1



User Interaction Engine Dataflow Diagram



Intercommunication Engine Dataflow Diagram