Coderecs

System Requirements Specification

Prepared by Harasees Singh, Harshpreet Singh Johar, Uttam Mittal and Abhinav Rawal
20103074 ,20103076 20103056 , 20103008

Version 0.1
September 12, 2022

Revision History

Name	Date	Reason for Changes	Version
Alpha Testing	12 Sept 2022	Initial Version	0.1

1. Introduction

1.1 Purpose

The purpose of this document is threefold:

1. To explain the process by which the preliminary requirements from Coderecs were analyzed and refined.
2. To state the requirements for the Coderecs in a way that allows tracing from the original (ambiguous, unrefined) form to the analyzed and refined form.
3. To show the analysis of the requirements (dependencies, issues resolved, how the requirements have been understood, etc.)

Therefore, this document is more than a requirements specification – it is also a statement of how the requirements were derived from the input, and how the requirements relate to each other.

1.2 Product Scope

The product is a software application which is accessed from a web browser and used by Competitive Programmers to build higher skill levels through a tailored learning experience.

1.3 Glossary

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document. Terms and concepts from the application domain are defined.

1.4 References // needs to be edited

● IEEE 830-1998

● CS6354 Project Documentation, by Bouchier, Brewster, Fischer, Herschbach, Nina

● Project submissions from CS6361, Summer 2006

1.5 Overview

This document is laid out in a modified IEEE 830-1998 style. The biggest difference from the
standard is the addition of section 4, which describes the process used to produce this document.
The following information is in this document:

- Section 2: General description of product, including user interface screenshots, dependency analysis, and enterprise requirements.
- Section 3: Functional & Non-functional requirements. Note that all non-functional requirements are grouped into one category, and not distributed among categories, as in IEEE830. Section 3 also covers use cases and deleted requirements.
- Section 4 describes the requirements analysis process.

2. General Description

Our solution will work by recommending problems to our users based on their codeforces profile. The solution will consist of two parts: a web application and a Deep Learning Model responsible for the problem recommendation engine. The solution will be deeply integrated with the codeforces profile of the user. Since codeforces is the most reputed site for competitive programming, our solution will recommend problems directly from the codeforces problemset based on the type of problems the user has solved in the past and the type of problems similarly rated programmers have solved. ‘Rating’, provided by codeforces, is a measure of the proficiency of a programmer in competitive programming and is generally an accurate measure for the same. Fetching the user data, which must be fed into the deep learning model, will be readily available using the various APIs provided by codeforces.com.

2.1 Product Perspective

Competitive Programming is a mind sport that gained popularity around 10 years ago. It involves tricky mathematical puzzles that require the amalgamation of algorithms, number theory and general logic to solve. It has a huge community which grows every year with college freshers entering the realm of coding from around the world. The most important factor that decides the leaderboard in programming contests is the ability to think out of the box. This is built over time with persistent practice.

Our product shall provide a way to beginners as well as intermediates an efficient way to practice and improve their skills.

2.1.1 System Interfaces // needs editing

As stated in section 2.1 the Scheduler system is a self contained system, relying on very little in
the way of external software interfaces. However, the system will require interfaces with the
installed the computer's hardware. The system is to be a web-enabled system, meaning that all user
interaction is done through a web browser. The System interfaces required on the system server
are the following:
o Network interface to a network with an internet connection
o Database connection to the mySQL database containing user and schedule data

2.1.2 User Interfaces

All user interfaces other than initial installation occur through a web page.

2.1.3 Hardware Interfaces

There are no hardware interfaces to this system.

2.1.4 Software Interfaces

The system will interface to an email system using SMTP.

// needs editing till here

2.2 Product Functions

Once the user signs up on our website, we will be extracting his/her details from codeforces and feeding it to the problem recommendation engine hosted on a remote server. Once the recommendations are ready they will be communicated to our frontend and displayed on the user's dashboard. We plan to provide 10 problem recommendations to our users and update them every 24 hours.

2.2.1 Enterprise Requirements

1. All users will be provided a set of 12 problems from codeforces which will be tailored according to his / her solving pattern, rating and problems solved by other people in a similar rating range.
2. Users will be able to track their progress and see their stats on the **coderecs/account page**.
3. Stats shall include a bar graph displaying number of problems solved by the user for each rating range from 800 to 3500 and a pie chart displaying the number of questions solved for each problem tag (like NT, data structures, math, flows, graphs, greedy etc.)

2.3 User Characteristics

There is only 1 type of user in this system. All users must authenticate through Google or Github during signup. All users will be entitled to a set of new problems which shall update after a fixed duration.

2.4 Constraints // eidt dis

There are a number of constraints which the system must abide by during development. The system must be developed within their bounds. These constraints dictate a number of the functional and nonfunctional requirements specified by this document. Others are because of a requirement specified to us by our customer. All are important to be aware of during the implementation of the software system.

- System is to be developed for distributed use as a web application. This will limit the ability for real time updates to the system.
- System is to be developed in Java through Servelets and JSP pages. Data must be stored in a relational database for quick queries and storage.
- Passwords must be sent and stored in encrypted form.
- Some users are authorized users while some are non-authorized users.
- Non-authorized users can not see other user's preference and exclusion sets.
- System must be robust enough to handle virtual meetings through teleconferencing, etc.
- System must handle rescheduling meetings with no outside input from initiator unless conflict arises
- System must be able to send email notifications to any common email server promptly and correctly
- Software Requirements Specification for Java Pet Store System 9 Keep user overhead to an absolute minimum.
- Anywhere the system can handle a decision itself, it must do so.

- Server-Client communication must be done over TCP connections
- Meetings can be rescheduled up to 24hrs prior to their current start time.

2.5 Assumptions and Dependencies

- The website can run on any modern browser like Google Chrome, Apple Safari, Microsoft Edge, Brave, Firefox etc.
- Any machine capable of running any modern browser and a stable internet connection can be expected to access coderecs without any problems
- Since coderecs uses APIs of codeforces.com, the users that signup for coderecs must have a codeforces handle
- The questions are fetched from codeforces.com therefore if for any reason the codeforces API ceases to work, coderecs won't be able to function properly and might require interference from the developers.
- The question set provided to a user can only be updated after a fixed duration.

3. Specific Requirements

3.1 External Interfaces

This section specifies the user interfaces to the system. All user interfacing is done through a web UI. The web pages are shown below.

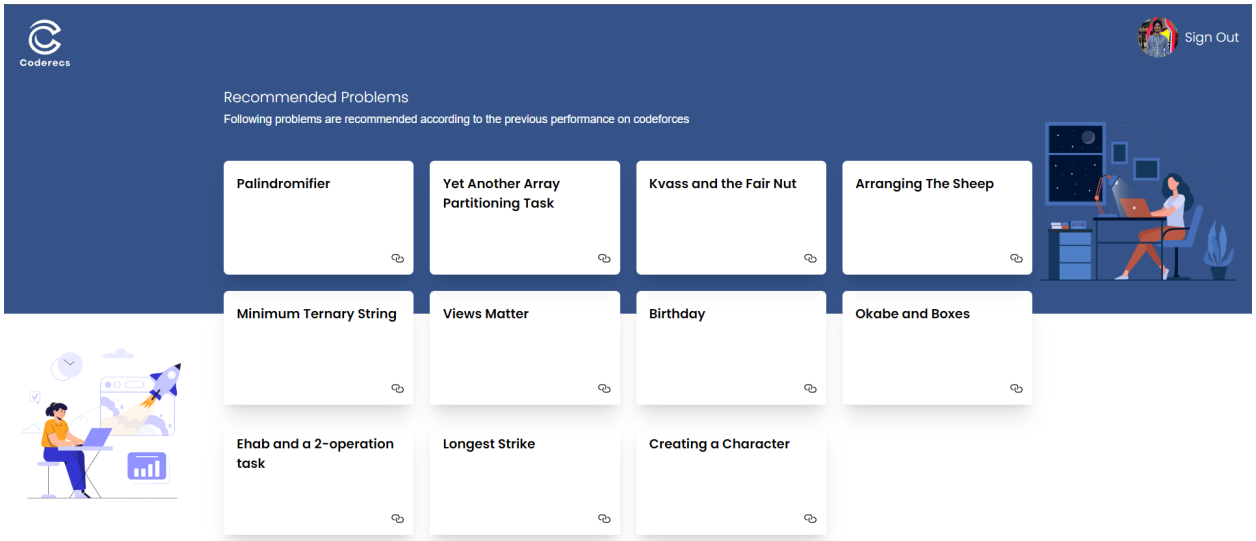


Figure 1: Dashboard

Here is a sample of the graphs displaying user stats.

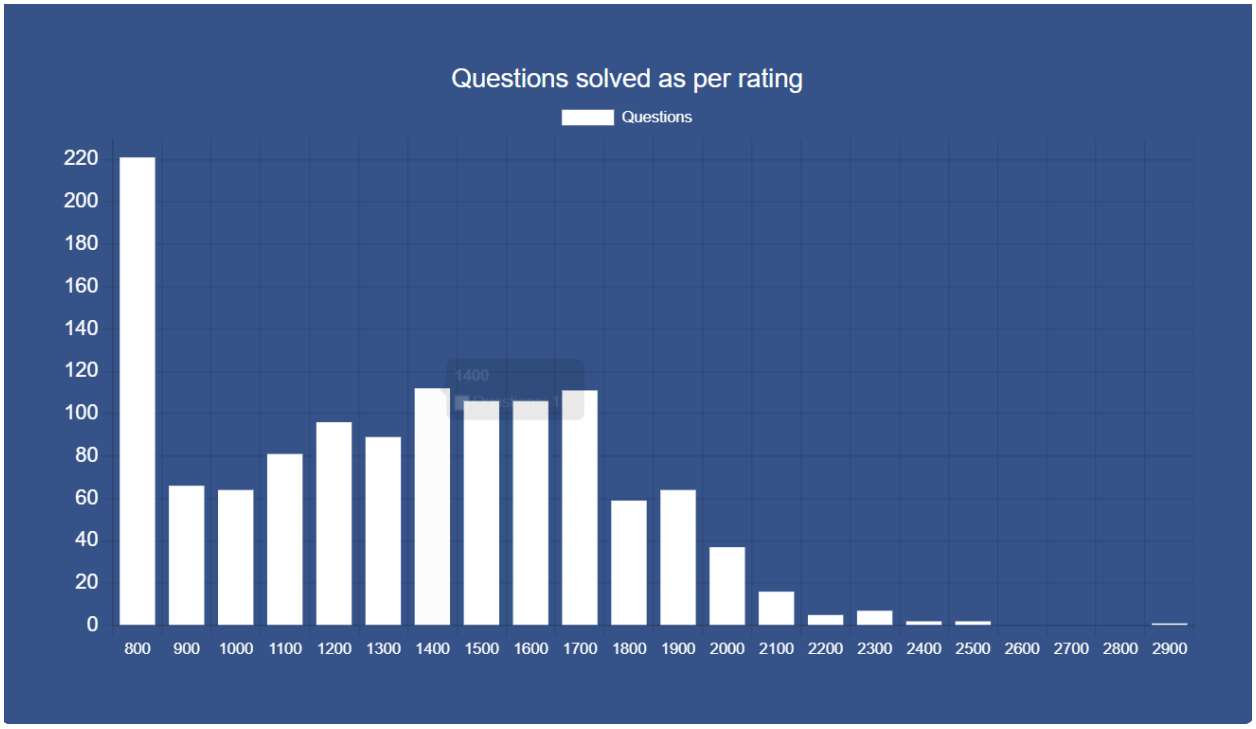


Figure 2: bar graph displaying number of questions solved corresponding to a particular rating

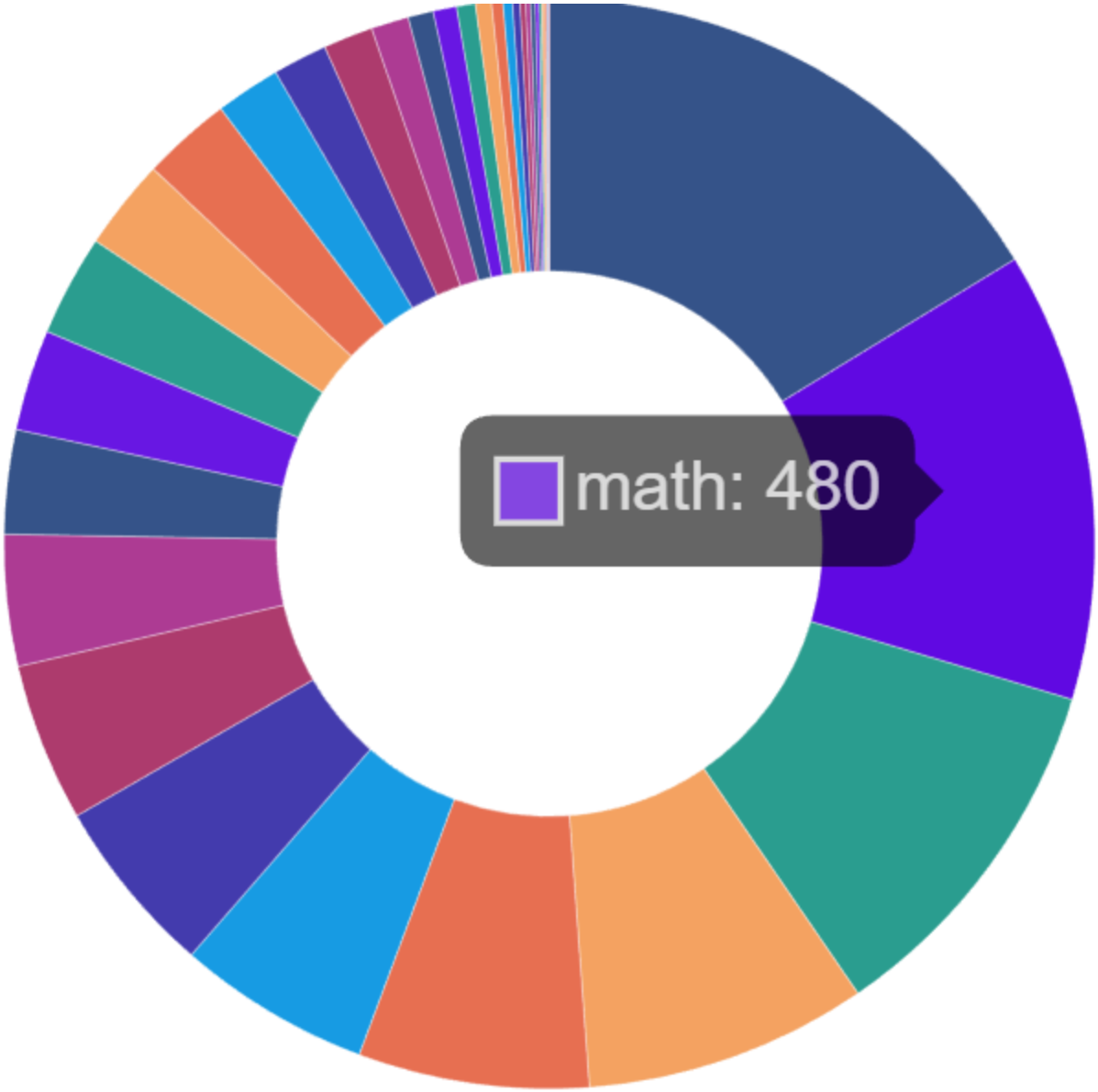


Figure 3: pie chart displaying the proportion of questions solved corresponding to a particular tag

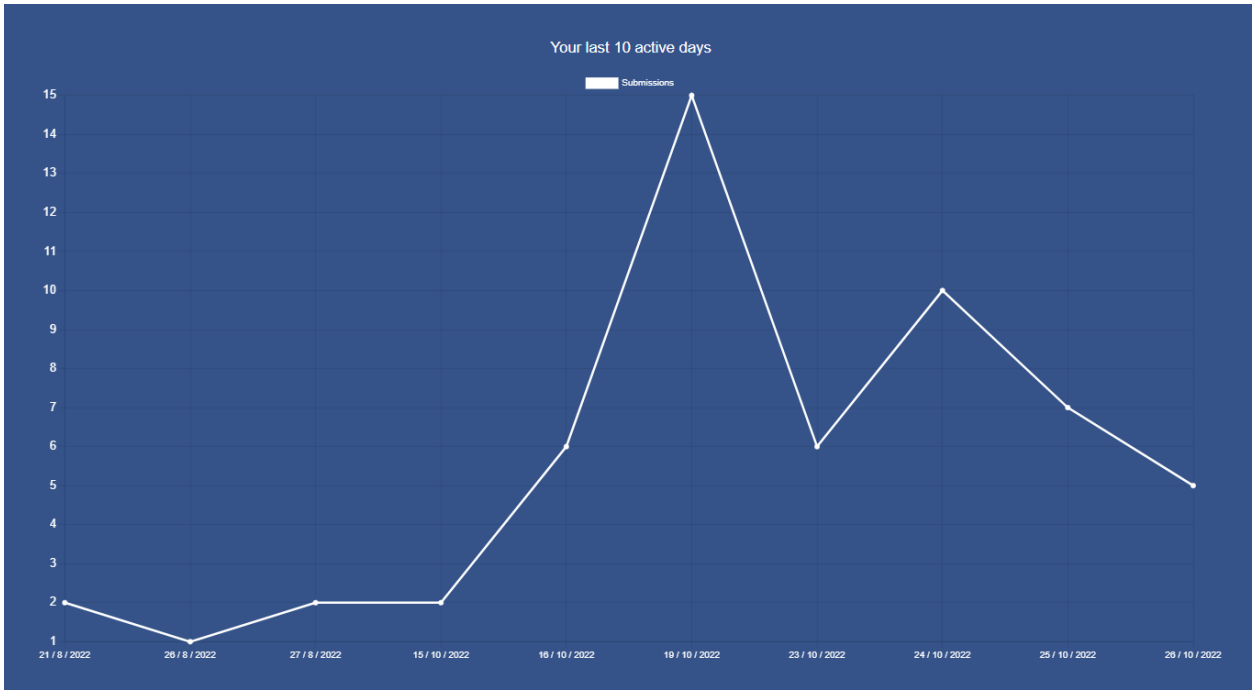


Figure 4: line chart displaying number of submissions made in the last 10 active days of the user.

3.2 Functions // edit dis

System functional requirements are specified by use cases and specific requirements. The use case helps understand system behavior, and the specific requirements extend the information from the use case.

3.2.1 System Functional Requirements

// don't know what to write dawg ☠️
// i don't think we have enough requirements

3.2.2 Nonfunctional Requirements

// something something

3.2.3 Deleted Requirements

// dis will be interesting to write 🤣

4. Requirements Analysis

4.1 Analysis Process

Each line in the Enterprise requirements was analyzed and recast where necessary to fix issues. After requirements were inserted, the requirements were analyzed for consistency, completeness, testability, correct categorization, and other requirements faults. Faults were corrected after consultation with subject & user representatives. The notes section of each requirement tells how it was modified from the original. Tracing was added to connect requirements.

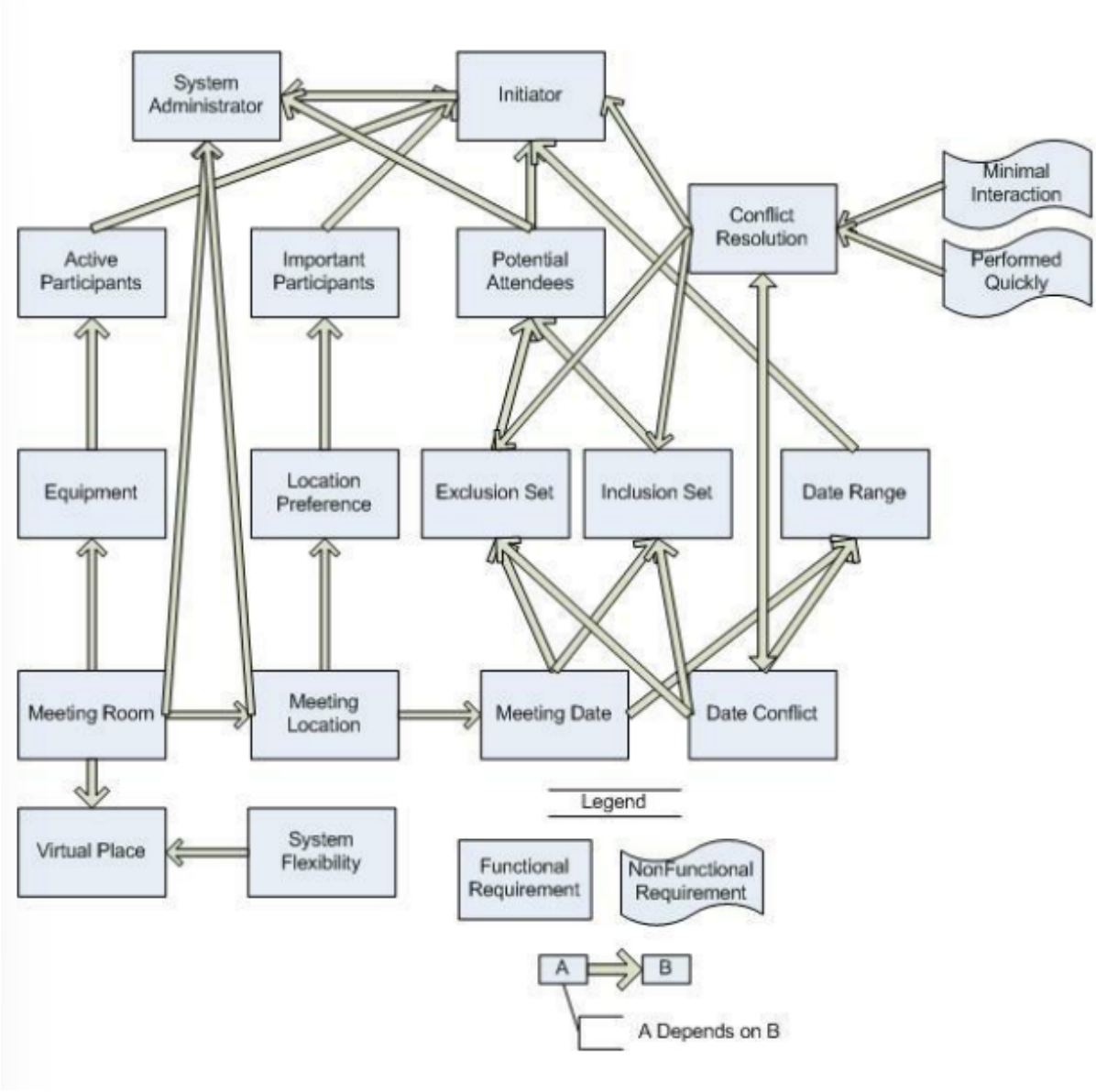
4.2 Dependency Analysis

Each concept in the Enterprise requirements was analyzed for meaning and dependencies and unresolved issues (dependencies could not be identified). The figure below shows which concepts or actions depend on each other in the enterprise requirements after the requirements refinement and issue resolution. A depends on B means without B there can be no A.

4.2.1 Enterprise Requirements Analysis

The refined (better understanding) dependency diagram for concepts in the system functional requirements is shown below

// insert a diagram showing dependencies

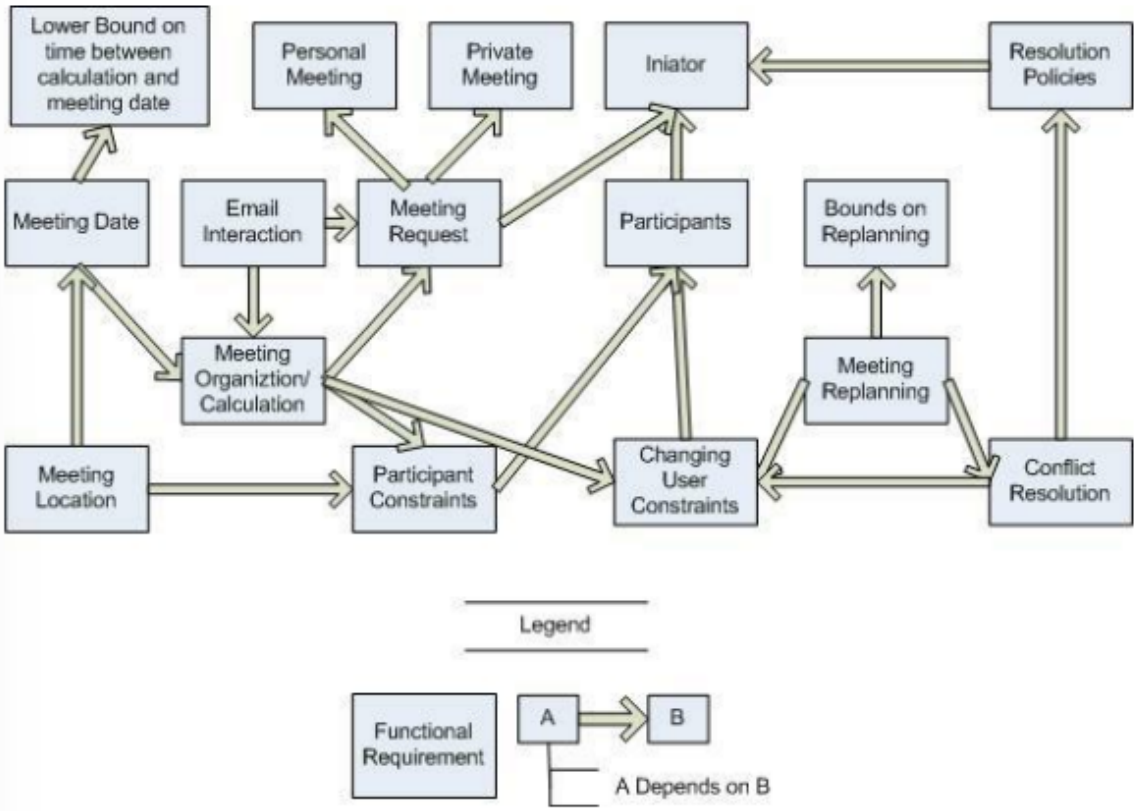


Example figure

4.2.2 System Functional Requirements Analysis

The refined (better understanding) dependency diagram for concepts in the system functional requirements is shown below

Figure 5: System Functional Dependencies (refined)

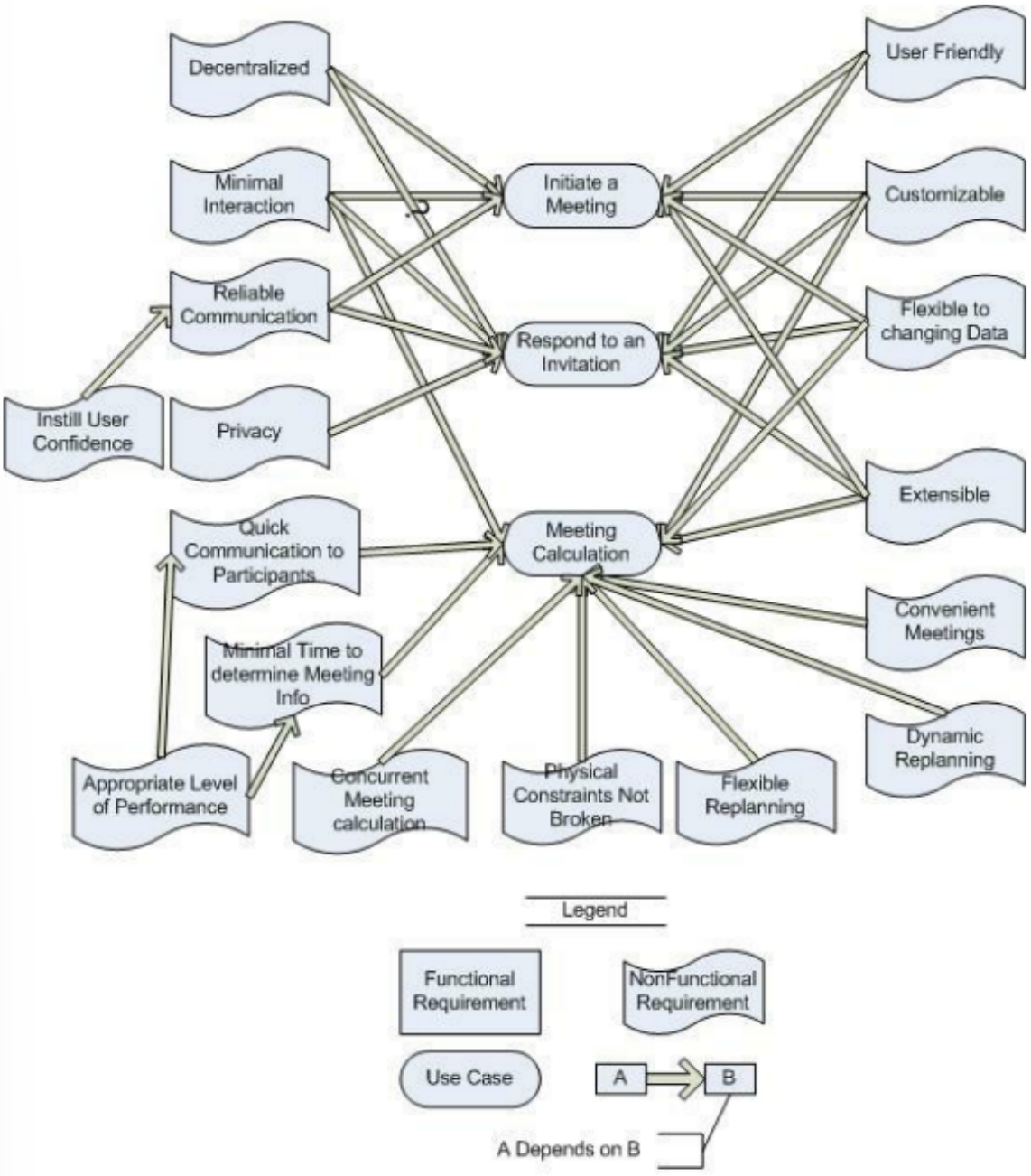


Example

4.2.3 System NonFunctional Requirements Analysis

The refined (better understanding) dependency diagram for concepts in the system nonfunctional requirements is shown below.

Figure 6: System non-functional dependencies (refined)



Example

4.4 Issues Raised to Coderecs

- Too much dependent on codeforces.com APIs
- No custom problems
- Problems should be updated on the request of the user
- Add a leaderboard

4.5 Original Requirements // to be edited