<Project Name>

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 4

2.4 Testability 4

2.5 Usability 4

3. Design Constraints 4

Supplementary Specification

# Introduction

An application that allows user authentication and registration and generates password for each username with given complexity.

It can be either a mobile or a web application.

Functional requirements:

* Define one type of user
* Authentication using username and password
* Process password generation using given user complexity

# Non-functional Requirements

Non-functional requirements:

* Validate all input data against invalid data before submitting the data and saving it into the data base
* Data will be stored in a relational database
* Use layered architectural pattern to organize the application
* Documentation

## Availability

Sistemul poate fi testat pe module mici, de la conectarea la baza de date si mapari pana la tranzactii si update uri.

In conditii optime, sistemul ar trebui sa functioneze in conditii optime.

## Performance

Sistemul dispune de o baza de data de dimensiuni reduse ca tabele si relatii. Am folosit un ORM pentru maparea entitatilor cee ace duce la performante ridicate.

## Security

Sistemul dispune de layere de Securitate in pachetul destinate interfetei grafice utilizator. Toate informatiile pe care le introduce utilizatorul sunt validate inainte de a fi procesare.

## Testability

## Usability

# Design Constraints

[This section needs to indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]