

---

**Gavril Luca**  
**30432**

---

**Tournament Supervisor System**  
**Supplementary Specification**

**Version <1.0>**

Tournament Supervisor System	Version: <1.0>
Supplementary Specification	Date: <10/03/2019>
Project_SupplementarySpecification	

## Revision History

Date	Version	Description	Author
<10/03/2019>	1.0	Initial Requirements System	Gavril Luca

Tournament Supervisor System	Version: <1.0>
Supplementary Specification	Date: <10/03/2019>
Project_SupplementarySpecification	

## 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	5
3.	Design Constraints	5

Tournament Supervisor System	Version: <1.0>
Supplementary Specification	Date: <10/03/2019>
Project_SupplementarySpecification	

# Supplementary Specification

## 1. Introduction

The Supplementary Specification captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

- Legal and regulatory requirements, including application standards.
- Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.
- Other requirements such as operating systems and environments, compatibility requirements, and design constraints.

## 2. Non-functional Requirements

### 2.1 Availability

The system is not expected to be used in urgent scenarios so we can afford a SLA of 99.5%. This translates into a yearly downtime of roughly 1 day and 19 hours, or a monthly downtime of 3 hours and 39 minutes. This time can be used to perform software updates, data compression and garbage collection.

### 2.2 Performance

In the worst scenario, the response time for the request of adding a team should not be longer than 30 seconds. The average response time, depending on the load of the system, should be 1 second.

### 2.3 Security

The system will be secured using https encrypted connections. Also we will demand user authentication and will not keep passwords in plain text. In regards to other user information, there is a restriction for an user to not be able to see the amount deposited by other users for their teams, although other information is available.

### 2.4 Testability

The business logic of the application must be tested independently from the user interface. We will employ V-Model testing as illustrated in **Error! Reference source not found..** We aim to have over 90% test coverage, through unit and integration tests. With respect to manual testing, the system will log all information that is not displayed in the user interface, so that the system is fully observable and testable.

Tournament Supervisor System	Version: <1.0>
Supplementary Specification	Date: <10/03/2019>
Project_SupplementarySpecification	

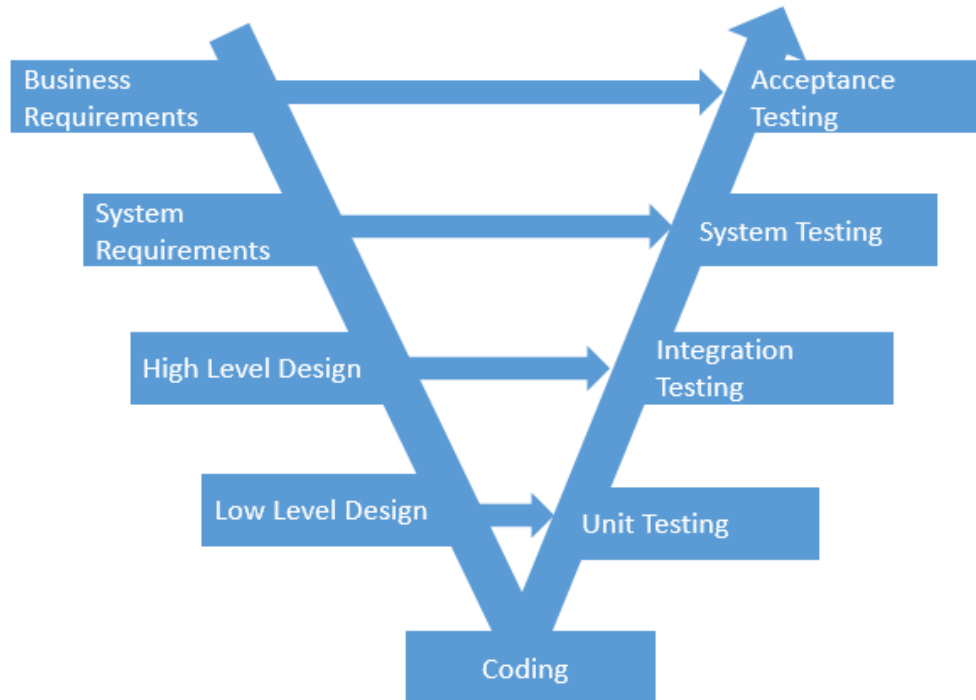


Figure 1. V-Model Testing

## 2.5 Usability

The user should be able to reach any desired goal in under 15 mouse clicks. The system state will be displayed for the organizer at all times, any terminal operation will prompt a confirmation dialog that describes the consequences of the action.

## 3. Design Constraints

The system is constrained to use Java 8 as implementation language. The software development process will be the Rational Unified Process (RUP), tailored to fit the team and the project. The conceptual architecture of the system will be a client server as illustrated in **Error! Reference source not found..** The required development tools are either Eclipse IDE or IntelliJ IDEA. In terms of libraries we will use: JavaFX, Hibernate, JDBC and GSON.

Tournament Supervisor System	Version: <1.0>
Supplementary Specification	Date: <10/03/2019>
Project_SupplementarySpecification	

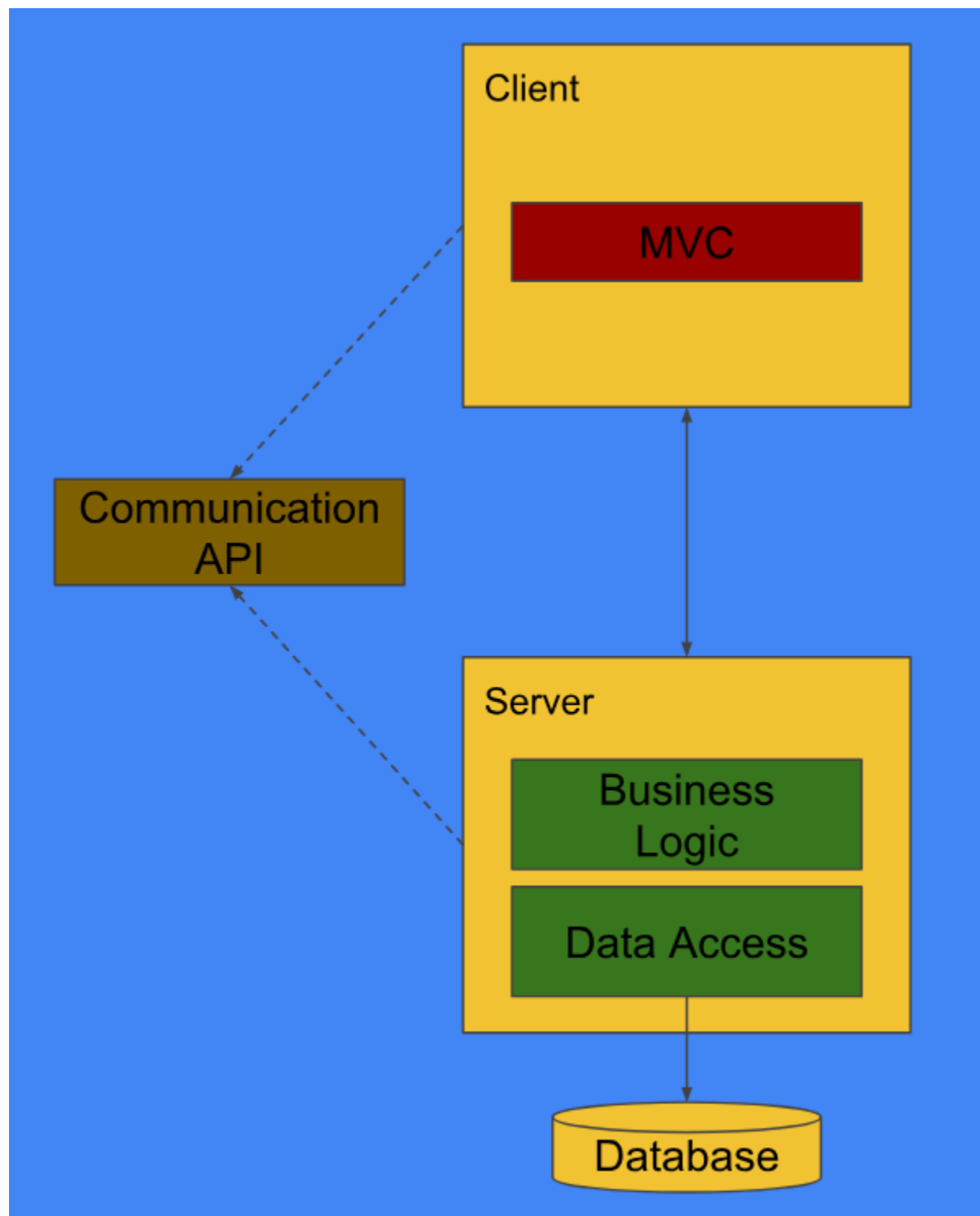


Figure 2:Conceptual Architecture