

---

**Vlad-Cristian Brînzăş**  
**30432**

---

**Composite Document**  
**Supplementary Specification**

**Version 1.0**

---

**Vlad-Cristian Brînzăș**  
**30432**

---

### **Revision History**

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
20/MAR/2019	1.0	Initial Requirements Statement	Vlad-Cristian Brinzas

## **Table of Contents**

<b>Introduction</b>	<b>4</b>
<b>Non-functional Requirements</b>	<b>4</b>
Availability	4
Performance	4
Security	4
Testability	4
Usability	5
<b>Design Constraints</b>	<b>5</b>

## **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<sup>1</sup> 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**

Performance is not an extremely factor for our system. For this reason we can allow a response time of up to 30 seconds for paragraph submissions and up to 5 seconds for displaying a story or user profile in the worst case scenario. The average response time, depending on the load of the system, should be 1 second or less.

#### **2.3 Security**

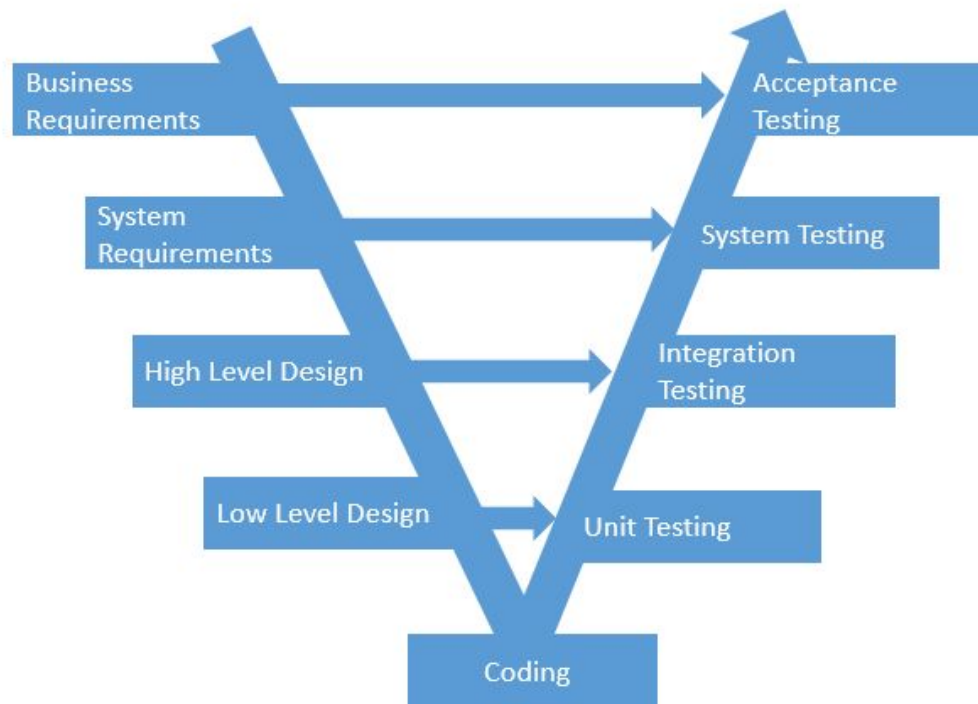
The system will be secured using https encrypted connections. Also we will demand user authentication for submitting paragraphs and will not keep passwords in plain text. Other user data will not be encrypted as we do not find it as being sensible information.

#### **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 Figure 1 V-Model Testing. 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.

---

<sup>1</sup> SLA = Service Level Agreement = Availability



**Figure 1 V-Model Testing**

### **2.5 Usability**

The user should be able to reach any desired goal in under 30 mouse clicks. Also, since the moderator might work under stress, any terminal operation such as suspending a user 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 Figure 2 Conceptual Architecture. The required development tools are either Eclipse IDE or IntelliJ IDEA. In terms of libraries we will use: JavaFX, Hibernate, JDBC and GSON.

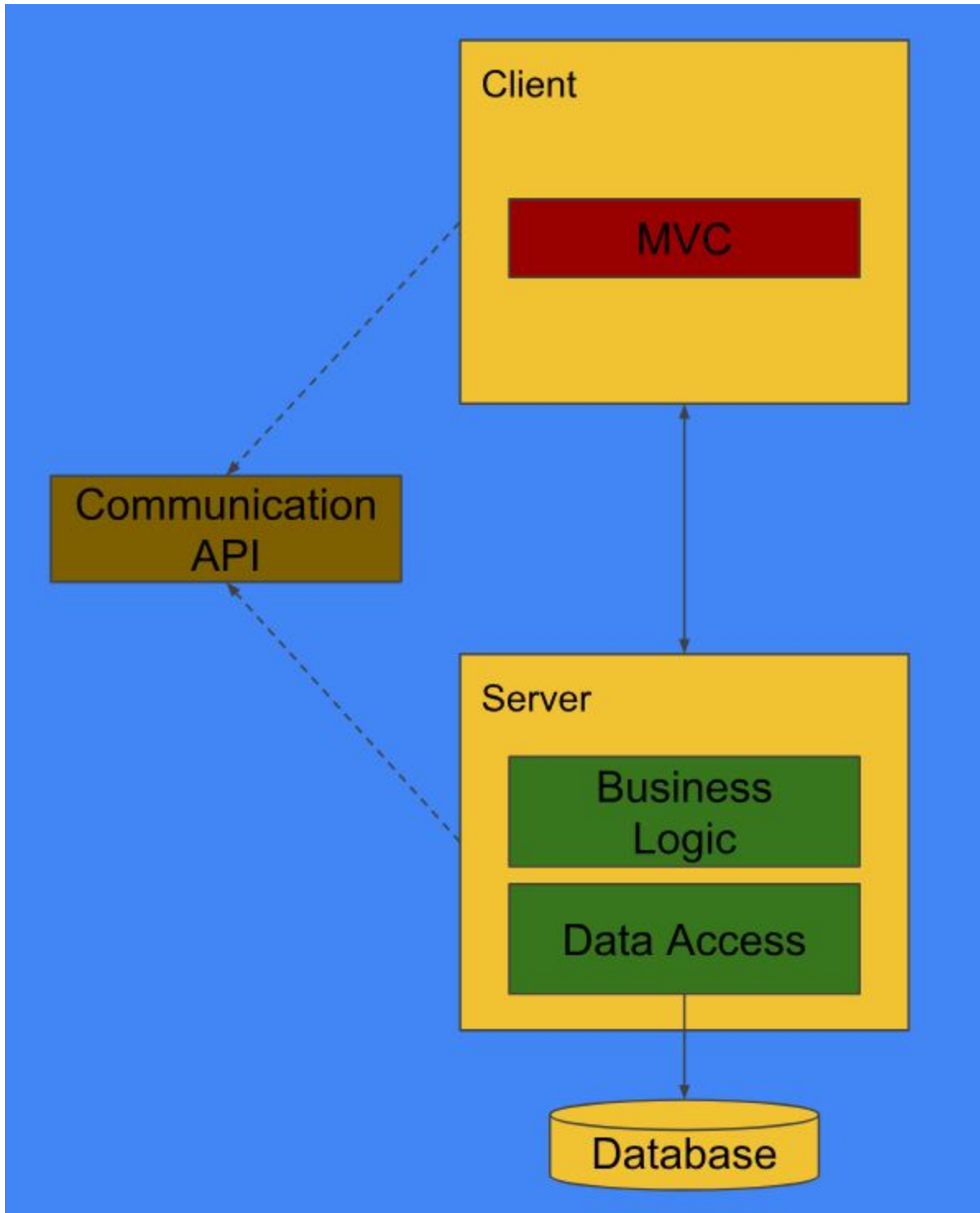


Figure 2 Conceptual Architecture