Saboteur Simulator

Version 2.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/03/2016 | 1.0 | Added few requirements and design constraints | Iulia Ungur |
| 27/05/2016 | 2.0 | Changed technology used and requirements | Iulia Ungur |
|  |  |  |  |
|  |  |  |  |

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

– version 1

# Introduction

The document provides an overview of the systems non-functional requirements as well as the projects design constraints. Such requirements include design and language decisions, availability, security etc.

# Non-functional Requirements

* Source of stimulus: the entity (human or another system) that generated the stimulus or event
* Stimulus: a condition that determines a reaction of the system
* Environment: the current condition of the system when the stimulus arrives
* Artifact: is a component that reacts to the stimulus. It may be the whole system or some pieces of it
* Response: the activity determined by the arrival of the stimulus
* Response measure: the quantifiable indication of the response
* Tactics

## Availability

* Definition: proportion of time the system is in a functioning condition
* Source of stimulus: server
* Stimulus: internet availability
* Environment: system freezes
* Artifact: data base
* Response: data base erases active games
* Response measure: system freezes
* Tactics: check internet connection

## Performance

* Definition: amount of useful work accomplished by the system
* Source of stimulus: user
* Stimulus: users connect to game play
* Environment: stand-by reaction – system awaits reaction from users
* Artifact: server, data-base, client
* Response: client announces information to server, server – db, db- server, server-client
* Response measure: move of user has been seen by others and db has been updated
* Tactics: client-server smooth connectivity

## Security

* Definition: degree of resistance to, or protection from, harm
* Source of stimulus: unidentified user
* Stimulus: db access
* Environment: system is in stand-by
* Artifact: server, db
* Response: server doesn’t permit access to db, encrypts the content and doesn’t allow direct change
* Response measure: error message
* Tactics: db isolation

## Testability

* Definition: degree to which the system supports testing in a given test context
* Source of stimulus: tester, business manager
* Stimulus: system normal flow of events
* Environment: system waits for commands and checks internet connection
* Artifact: Server
* Response: server retrieves necessary information for instruction execution
* Response measure: db updates, server gives desired instructions and updates page
* Tactics: code review, automated tests, situation tests

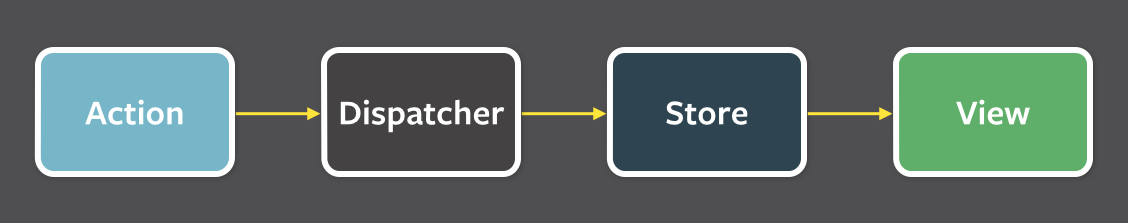
## Usability

* Definition: degree to which the system can be used by specified consumers to achieve quantified objectives
* Source of stimulus: unexperienced user
* Stimulus: log in and registration
* Environment: system waits for identifiers data
* Artifact: db, server
* Response: if details correct and connectivity available, system logs in, in desired page
* Response measure: db has been updated with new user, and lobby /chat is presented
* Tactics: db availability, server detail verification, ease-of-use

# Design Constraints

Design decisions:

* Software language and tools:
* server: JavaScript (Typescript) – used: MongoDB, Express, Node.js;
* client: JavaScript, HTML, CSS – used: Angular 2 beta 11
* Development tools: NPM (Node Package Manager), TSC (Typescript-Compiler), Gulp (automation of built and compilation process) [[1]](https://www.npmjs.com/package/gulp)
* Architectural and design constraints:
* Client-Server request based architecture for communication – using XHR, with JSON messages



* Angular 2 using Flux Architecture (see fig below) [[2]](http://blog.jhades.org/angular-2-application-architecture-building-flux-like-apps-using-redux-and-immutable-js-js/)

– version 2

# Introduction

The document provides an overview of the systems non-functional requirements as well as the projects design constraints. Such requirements include design and language decisions, availability, security etc.

# Non-functional Requirements

* Source of stimulus: the entity (human or another system) that generated the stimulus or event
* Stimulus: a condition that determines a reaction of the system
* Environment: the current condition of the system when the stimulus arrives
* Artifact: is a component that reacts to the stimulus. It may be the whole system or some pieces of it
* Response: the activity determined by the arrival of the stimulus
* Response measure: the quantifiable indication of the response
* Tactics

## Availability

* Definition: proportion of time the system is in a functioning condition
* Source of stimulus: server
* Stimulus: server availability
* Environment: system freezes
* Artifact: server
* Response: server erases active games
* Response measure: system freezes
* Tactics: check server connection

## Performance

* Definition: amount of useful work accomplished by the system
* Source of stimulus: user
* Stimulus: users connect to game play
* Environment: stand-by reaction – system awaits reaction from users
* Artifact: server, client
* Response: client announces information to server, server-client
* Response measure: move of user has been seen by others
* Tactics: client-server smooth connectivity

## Security

* Definition: degree of resistance to, or protection from, harm
* Source of stimulus: unidentified user
* Stimulus: lobby access
* Environment: system is in stand-by
* Artifact: server
* Response: server permits access to anyone who is registered in the system (joined a game)
* Response measure: error message
* Tactics: periodical check of correct data

## Testability

* Definition: degree to which the system supports testing in a given test context
* Source of stimulus: tester, business manager
* Stimulus: system normal flow of events
* Environment: system waits for commands
* Artifact: Server
* Response: server retrieves necessary information for instruction execution
* Response measure: server gives desired instructions and updates game or lobby
* Tactics: code review, automated tests, situation tests

## Usability

* Definition: degree to which the system can be used by specified consumers to achieve quantified objectives
* Source of stimulus: unexperienced user
* Stimulus: makes a move
* Environment: system waits for identifiers data
* Artifact: server
* Response: if details correct, system responds
* Response measure: game updates map
* Tactics: server detail verification, ease-of-use

# Design Constraints

Design decisions:

* Software language and tools:
* Server: C#
* Client: C#
* Development tools: VS 2015, NetworkComms
* Architectural and design constraints:
* Client-Server request based architecture for communication
* Observer Pattern for lobby and game refresh