

# Business Simulation Software (BSS)

Software requirement specification

Technical specification



## Contents

1. Overview.....	2
2. Features and Functionalities.....	3
2.1. Backend system.....	3
2.2. Frontend system.....	4
3. User stories.....	5
3.1. Creating new scenarios (role: administrator).....	5
3.2. Performing simulation (role: user).....	6
3.3. Reviewing output (role: professor).....	6
4. Screens and navigation.....	6
4.1. Screen 1: User login screen (frontend).....	7
4.2. Screen 2: User viewing available simulations (frontend).....	7
4.3. Screen 3: User performing simulation (frontend).....	8
4.4. Screen 4: Viewing completed simulations (frontend).....	9
4.5. Screen 5: Administrator login (backend).....	10
4.6. Screen 6: Professor viewing the simulations (backend).....	10
4.7. Screen 7: System administration (backend).....	10
5. System architecture.....	10
6. Questions and scenarios.....	12
7. Documentation and testing.....	13
8. Support.....	13
9. Software source code.....	14
10. Index of abbreviations and acronyms.....	15

# 1. Overview

The Business Simulation Software (BSS), which will be implemented as part of the SPIDE project, will represent a web-based ICT tool which will enhance the innovative and creative capacities of students during their course lectures in subjects in the fields of entrepreneurship and management.

This simulation software will be filled with scenarios, each scenario to be resolved over the period of 45~75 minutes. Through the software, the students will be presented with a certain environment, with multiple possible business decisions to be made. Each choice leads to other decisions, until the student reaches the final outcome, which can be Success or Failure. Each step is graded and the students are finishing the simulation with a concrete number of grades.

While making the exercise, students will record the outcome and the professor can add additional points, depending of the methodology and approach of the students, in solving the particular business simulation.

The simulations can be worked as individuals or as groups.

The motivation for creating such a software, is in relation with the objectives and goals of the SPIDE project. BSS was created with intent to:

- Provide Business “flight simulation” for students;
- Help students learn from real-life cases;
- Increase entrepreneurial skills;
- Teamwork and problem analysis;
- Adding innovative component to existing systems and curricula.

With the BSS platform, the SPIDE project team is expecting to inspire multiple aspects of the business and entrepreneurship skills of the students, among which:

- Strategic thinking;
- Decision making;
- Different approaches to problem solving;
- Financial analysis on real-life cases;
- Market analysis on real-life cases;
- Managing everyday operations;
- Self-determination, motivation and enthusiasm;
- Teamwork and leadership.

The resulting software should be able to provide:

- Internal online software system;
- Tested quality of the software;
- availability for gathering feedbacks from users of the software;
- Holding a minimum of 10 real-life scenarios;
- Ready for usage package, which will later be involved into existing curricula on the partnering institutions.
- The system will provide grading output and the best team throughout the

summer semester of 2017/2018 will visit the StudentsHub event in Slovenia.

The BSS will present an innovative concept in the educational process to current students that will merge the utilization of practical experience and ICT tools, in order to eventually match labor market needs of the business sector. Despite presenting a simulated business environment to students, the BSS will be a learning experience, allowing for students to see, correct, and learn from their mistakes throughout the process.

The Business Simulation Software which will be used by the entrepreneurship students on the classes in the four partnering institutions. These activities will take part in the third and fourth semester of the project duration. Final delivery date for the software, including testing and documentation, is 4 (four) months after the start of the project, calculated from the date the contract is signed with the chosen software development company.

## **2. Features and Functionalities**

The BSS software package should have a divided architecture of a backend and frontend system, residing on a MVC (Model view control) architecture. The frontend and the backend system, should communicate through RESTful or REST API and can reside on a single or multiple computer systems.

Throughout this document, simulations are also referred to as scenarios. They both have the same meaning, though scenario is used when a template is created on the backend, while simulation is referred to when students are performing the particular scenario through the frontend of the application.

### **2.1. Backend system**

The backend system should communicate with the database, be responsible for management of the model, hold the control logic and present the required data to the frontend system through a HTTP RESTful API.

The backend system should be used by system administrators and not by the end users (performing the actual simulations) directly. Its purpose is management of the system, creating/editing users and scenarios.

The backend system should be able to:

- Manage the database directly, without the necessity to access the database directly, in order to modify its tables and columns.
- Communicate with MySQL or PostgreSQL databases.
- Have backend web-based management system for adding the data directly, such as creating users, scenarios and options.
- Provide a listing of all available users, ability to add or modify records.
- Provide a list of all available scenarios, ability to add or modify records.

- Provide a list of all available options, decisions and metrics for a particular scenario and the ability to add or modify records.
- Provide a comprehensive view of a scenario, including all levels and all possible decisions on a single screen.
- The ability to batch-upload list of users to the system.
- Provide a list of all completed simulations, score, notes and information about the team or individual who participated in it.
- Export a detailed list of all completed simulations.
- Have built-in authentication system.
- Handle password reset for a user, by an administrator through the backend side, sending new password or link for password change to the user on e-mail.
- Reset the state of the simulation, if it is to be attempted again by the same user or group of users.
- One simulation can be attempted multiple time by the same user or group of users.
- One simulation can be attempted either by a user individually or by group of users, choosing to do group simulation. If group simulation is performed, the system must record the role the user has in that particular project (role such as: production manager, finance manager, etc.)

## **2.2. Frontend system**

The frontend system should communicate with REST API with the backend system, using a token and IP address restriction, in order to authenticate against it. Not all functionalities of the backend will be implemented on the frontend side, such as creating scenarios and reporting. It's main goal is to present the simulation data in a clear and convenient way to the end users.

The frontend system will be used directly by the end users of the system and it is not intended for system administration, which should be handled through the backend of the system.

The frontend system should be able to:

- Have responsive layout and be viewable on regular computer, mobile and tablet devices.
- Use authentication for the end users, which will be checked against the backend system, through the API.
- Provide users profile page where they can change their personal data and passwords.
- Provide the user with list with available simulations.
- Provide the user with list with completed simulations and their score.
- Provide the user an option to start a simulation, retrieve the data from the backend system and present it to the user.
- Remember the progress of the user, so that if the simulation is stopped (or connection lost by the end user) it can be resumed later, from the

- same stage.
- If a simulation is preformed by a group of users, when each of them logs on to the frontend system and enters the particular simulation - all of them must see the same screen. If a decision is inputed by one of the user, it must be viewable and automatically redirected to the next option, on all frontend screens, for all users participating in the particular simulation.

The frontend system must be built with HTML/HTML5 and/or JavaScript components and not use Flash, in order to be available on desktop, mobile and tablet devices.

### 3. User stories

The purpose of this section is to explain the flow of information and operations, which the users must undertake in order to perform some of the core functions on the BSS platform.

The system must have three different user types:

- **Administrator** - be able to modify all data on the backend side. Create new simulations.
- **Professor** - login to the backend of the system, view the performed simulations and options.
- **Student** - login to the frontend of the system, view and perform simulations.

Users should be created from the backend of the system, either directly or by uploading a predefined CSV file with information about the uses. If users are uploaded from the CSV file, they should automatically be notified by e-mail, about their username (e-mail address) and password (generated automatically or link so that they can access and change their password).

#### 3.1. Creating new scenarios (role: administrator)

New scenarios should be crated through the backend of the system, as administrator user should be able to:

- Logon to the backend system with username and password.
- View a list of available scenarios.
- Click on a control to add new scenario.
- Enter basic scenario info (as described in section 5).
- Enter the desired result or final outcome for that scenario.
- Create the first level set of options (decisions available at the first level).
- When entering into a particular option, create an arbitrary number of sub-options or sub-decisions.
- For every sub-option or sub-decision, choose whether if results in final outcome or leads to another level or another decision list.
- Mark the simulation as available, so it should start to appear in the

frontend system, to the end users.

### **3.2. Performing simulation (role: user)**

Simulations should be performed by end users, through the frontend of the system, as the users should be able to:

- Logon to the frontend system with username and password.
- View a list of available simulations (including the simulations they have chosen to participate in previously).
- If a simulation is not completed, the user must finish the previous simulation, before starting a new one.
- The user starts a new simulation and is presented with the initial data, metrics and the initial set of decisions to be made.
- The user should choose a particular option and input notes about why that particular choice was made. Same goes with all other nested options, until a chosen option results with the final outcome of the scenario, which can be either Success or Failure.
- The user can download in PDF the simulation record, with the made choices, notes and metrics data.

### **3.3. Reviewing output (role: professor)**

Simulations review should be performed by the professor, through the backend of the system, as the users should be able to:

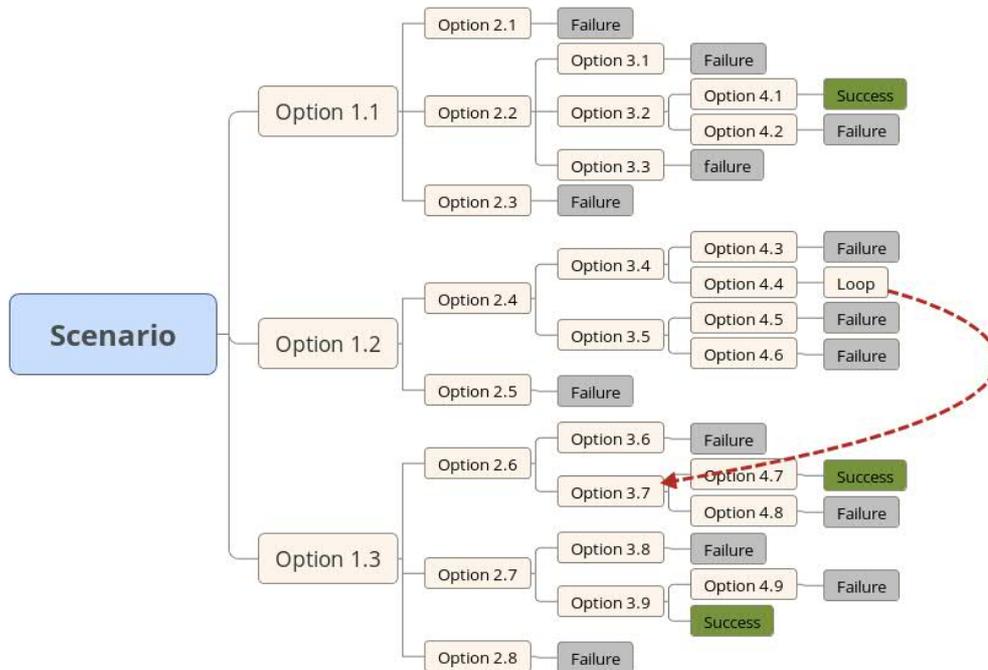
- Logon to the backend system with username and password.
- Navigate through the institutions and courses, in order to get to the list of simulations performed on that particular course.
- Opening the performed simulation, viewing all the choices and notes made by the students.
- If the professor (as a backend user) is associated with that institution and course, which should be selected on the backend of the system, for that particular professor - he can add additional points on behalf of the notes/methodology/approach made by the students. The amount of points which can be added by the professor should not exceed 20% of the total score accomplished by the student or group of students, for that particular simulation.
- The professor can export all of the simulations for a particular course or institution in csv form.

## **4. Screens and navigation**

This section explains the screens which the users should see when using the system. The concrete design of the screens is left to the software development company and each of the developed screens must meet minimum the

requirements listed in this section.

On the following diagram, there is a breakdown of one scenario and all of its available options and outcomes. Through the backend of the BSS system, all options are created individually, as every option is linked to another set of options or an outcome. On the frontend side, when user performs the simulations, they can view all made choices on a single screen, together with the metrics which result from their decisions.



Picture 1: Diagram of all available options for a particular scenario

#### 4.1. Screen 1: User login screen (frontend)

This screen should contain:

- The name of the BSS platform.
- The logo of the SPIDE project.
- Field with username.
- Field with passwords.
- Submit control.
- Control for "Forgot my password", so the user can receive new password or link for changing his password on e-mail.

#### 4.2. Screen 2: User viewing available simulations (frontend)

This screen should contain:

- The name of the BSS platform and SPIDE project.
- Main navigation menu on the left of the screen.

- Sign out control on the bottom of the left navigation menu.
- On mobile devices, the navigation menu should be hidden and accessible when clicking on the menu control.
- Table with list of available simulations.
- Information whether the simulation has been performed previously, with number of previous attempts.
- Information about the top score of the particular simulations (from all previous attempts) and the attempt in which the top score has been accomplished.
- If a simulation is marked as available (from the backend side), control for "Start simulation" should be available to the user.
- If a simulation is not marked as available, user can see it in the listing of completed simulations (with information about number of attempts and scores), but shouldn't have the control to start the simulation again. Then the simulation is not present on the "available simulations" screen, but on the "completed simulations" screen.
- If a user is already in the middle of performing a simulation or is assigned to a particular simulation (as a member of group), instead of "Start simulation", he should see a control "Continue simulation".
- When the user clicks "Start simulation" or "Continue simulation" he is redirected to the next screen - performing the presentation.

### **4.3. Screen 3: User performing simulation (frontend)**

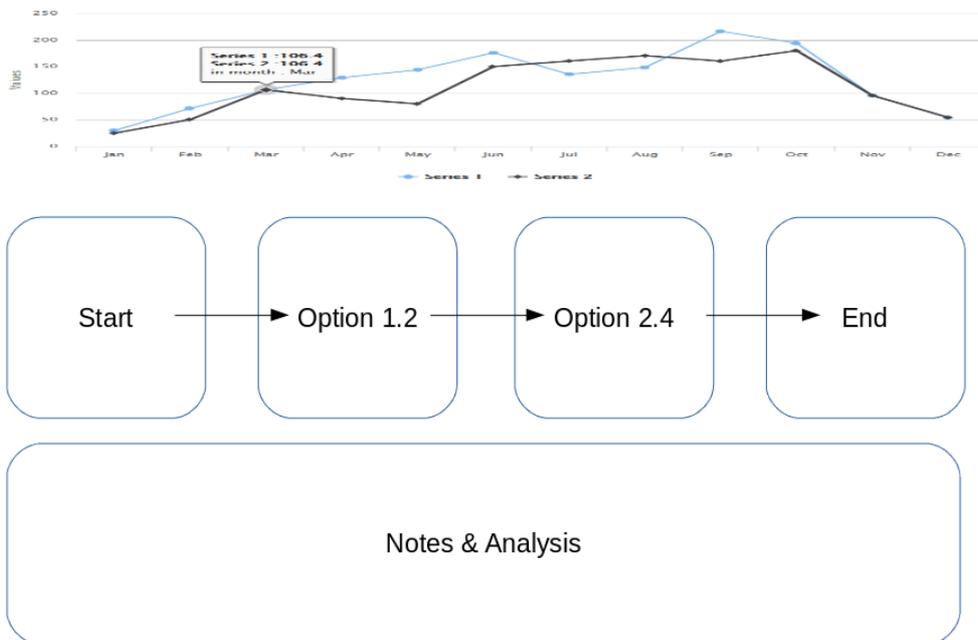
This screen should contain:

- The name of the BSS platform and SPIDE project.
- The name of the scenario on the top of the screen.
- The navigation menu on the left should be auto-hidden, with the ability to retrieve it through a control on the top left of the screen.
- In the top of the screen there should be a graph with the metrics from the particular simulation, with data to the point the user has reached in the present attempt. If the user is at the first level of available choices, only the data until that point is displayed on the top graph.
- Below the graph, there is the list of options, until the point the user has reached in the present attempt. The current list of available options, on the level the user has reached in the present attempt are displayed on the screen and the user can click on a particular option.
- When the user makes a decision and clicks on an available option, a pop-up screen is brought up where the user can add mandatory notes about why that decision has been made.
- After the decision is made, the user is presented with the outcome of the decision, which may lead either to a final outcome or another set of available choices.
- In the bottom of the screen, all of the notes made by the user, while performing the simulation can be seen.
- This screen is presented to all participants, if group simulation is being performed, dynamically reloading after a choice is made by one of the

users.

- When user reaches a final outcome, the simulation is marked as completed in the backend of the system and the user cannot continue or make different set of choices. He can only re-start the simulation again, if it is still in the list of available simulations.

Example of the design for this screen can be seen on the following image.



Picture 2: Design of the frontend screen for the end user, while performing a simulation

#### 4.4. Screen 4: Viewing completed simulations (frontend)

This screen should contain:

- The name of the BSS platform and SPIDE project.
- Through the navigation menu on the left, the user can navigate to the screen for completed simulations.
- Information about the number of attempts and previous top score can be seen in the table containing previously attempted simulations.
- User can click on the simulation and be presented with the final "presentation screen" (as described in section 4.3).
- User can also click on a control to export the completed simulation in PDF.
- The PDF export should contain all of the information of the attempt, including the metrics graph, decisions made and notes made by the student.
- If the professor has decided to grant extra points to the particular simulation, it should be seen in the listing of the completed simulations, the final presentation screen or the PDF exported document.

#### **4.5. Screen 5: Administrator login (backend)**

This screen should contain:

- The name of the BSS platform.
- The logo of the SPIDE project.
- Field with username.
- Field with passwords.
- Submit control.
- Control for "Forgot my password", so the user can receive new password or link for changing his password on e-mail.

#### **4.6. Screen 6: Professor viewing the simulations (backend)**

This screen should contain:

- The name of the BSS platform and SPIDE project.
- List of institutions, navigating to list of courses and list of completed or ongoing presentations.
- When the professor clicks on a completed presentation, he should be able to see all data regarding the made decisions, notes and metrics for that particular presentation.
- If a presentation is completed, the professor can also choose to grant additional points to the particular simulation, on behalf of the notes or explanations made by the students. This score must not exceed 20% of the already accomplished score, by the students.

#### **4.7. Screen 7: System administration (backend)**

This screen should contain:

- The name of the BSS platform and SPIDE project.
- All available models, as described in section 5.
- Simple tables for searching, adding and editing records for the particular models:
  - Institutions;
  - Courses;
  - Users;
  - Scenarios;
  - Scenarios -> Options;
- Simple tables for viewing records for:
  - Simulations;

### **5. System architecture**

The BSS system must work on a GNU/Linux platform, which includes both the frontend and the backend of the system. The database used must be either MySQL or PostgreSQL and should be installed on the same system as the backend system.

The purpose of section is to explain the database model of the system and the structure of each of the particular components.

When creating new simulations or scenarios, each of them is assigned to a particular institution or course. When a professor logs on to the backend of the system, he can view all of the simulations performed per institution and course. However, the professor can review and update scores only to the simulations made on the Institution and Course, the professor is associated with.

Here is a list of models and the data which should be kept for each of them:

- Institution:
  - Name
  - Address
  - E-Mail
- Course:
  - Name
  - Professor (foreign key to User of type Professor)
  -
- User:
  - Name
  - E-Mail address (login name)
  - User type (Administrator, Professor, Student)
  - Picture
- Scenario:
  - Name
  - Description
  - Goal
  - Challenges
  - Opportunities
  - Duration
  - Roles (available roles, to be assigned to particular students, for group participation in the simulations)
  - Start metrics
- Option
  - Name
  - Offset (level from the start of the simulation)
  - Pros
  - Cons
  - Opportunities
  - Risks:
- Decision (adding notes):
  - Motivation (entered through the frontend by the students)
  - Risks (entered through the frontend by the students)
  - Opportunities (entered through the frontend by the students)

- Score (entered through the backend by the administrator, not viewable by the students on the frontend)
- Metrics
  - List of available metrics, to be included in scenarios.

Explanation about some of the core relations in the data model of the system:

- Institution - course (one-to-many);
- Professor - course (one-to-many);
- Scenario - source (many-to-many);
- Scenario - options (one-to-many);
- Scenario - metrics (many-to-many);
- User - Scenario (many-to-many);

## 6. Questions and scenarios

The BSS platform will consist of problems which real-life companies, entrepreneurs and decision makers faced. The students during the lecture will be divided into different groups, and each group will be assigned to a company. The professor will be responsible for explaining the rules to the students, guiding them through the process and inputting all information in the software. As the BSS platform is implemented into onto the entrepreneurship classes, activities have also been planned for maintenance and quality assurance of its operations.

Mostly expected scenarios to be created are involving startup companies or development of new products. It is expected that:

- Scenarios will be intended to be competitive;
- Scenarios should be Interactive, students will make the selections themselves;
- Scenarios should be generic, non-industry specific.
- Scenarios should be team oriented;
- Scenarios would be deterministic, with exact options and selections.
- Simulations should be performed or handled over a period of one class time (between 45 and 75 minutes).

The following main industries are expected to feature in the newly created scenarios:

- Research and development;
- Human resources;
- Marketing;
- Production and manufacturing;
- Finance;

## **7. Documentation and testing**

The Business Simulation Software (BSS) needs to be accessible by multiple end users, on different platforms, simultaneously and from different geographical location, both on the frontend and backend side. It is therefore imperative that the software goes through processes of both automated and internal testing, before it goes into production and is used from the SPIDE partner institutions.

While the internal testing will be performed by project team members of the SPIDE project, the automated quality assurance testing must be performed by the company responsible for development and implementation of the software platform. The automated testings should include:

- Testing on different devices and commonly used screen resolutions;
- Stress-testing with up-to 1000 simultaneous users;
- Security testing for potential security weaknesses and vulnerabilities;
- Testing basic scenarios:
  - user login (frontend and backend),
  - scenario list (backend),
  - simulations list (frontend),
  - create scenario (backend),
  - create option (backend),
  - perform decision (frontend),
  - retrieve metrics (frontend).

The complete delivery of the BSS software system should include:

- Documentation for software developers, used frameworks, source code overview and structure, instructions for future development of the platform.
- User manuals for: Administrators, Professors, Students, for most common operations.
- Instructions for installation of the system
- Instructions for maintenance of the system, backup of the database and the frontend and backend applications.
- Instructions or re-installing the system on different computers with restoring an existing backup of the database.

## **8. Support**

The software should be delivered in 4 months after signing the contract with the software development company or until 31.12.2017.

Within that period, apart from the complete installation of the software the company responsible for development and implementation of BSS should also deliver:

- Quality assurance testing of the backend and the frontend of the system.
- Training performed to members of the SPIDE team, which will be responsible for administration of the system.
- Documentation for using the system, on behalf of the three user types.
- Instructions for installation of the system, backup of the database and re-installing the system on different computers with restoring an existing backup of the database.
- Source code of both the frontend and the backend of the application.

Support for the software must be included until the completion of the SPIDE project, until 31.12.2018. The support should include:

- Bugs and issues which were detected during the operations of the software, in the period until the support is included.
- Questions and information regarding regular software operations and maintenance.

The response time for the support, after reporting the problem on e-mail by the SPIDE team responsible for BSS administration, should be:

- 1 working day, for critical problems;
- 10 working days for non-critical bugs, problems or questions.

## **9. Software source code**

All of the software source code, delivered in the frontend and the backend of the system should be delivered to the project. This includes both open and closed source components, which will be used exclusively by the partnering institutions and for the purposes of the BSS platform.

The SPIDE project holds all rights on the software source code, to use it and modify it, for the needs of this particular project and the BSS software platform.

The software can be both open source and closed source system. However, the source code must be delivered to the Some components of the system can be closed source, but SPIDE must ho

## 10. Index of abbreviations and acronyms

- **BSS** - Business simulation system.
- **Simulation** - Running simulation, by students.
- **Scenario** - Scenario template, created by professor.
- **Backend system** - Software component which serves data from the database to front-end system, implementing logic and control. Accessible from the teaching staff only.
- **Frontend system** - Web site accessible by students.
- **REST API** - RESTful web services for interface with applications.
- **Decision** - Selection of a choice, made by students in a simulation.
- **Choice/Option** - Possible choice in a scenario, created by professor.
- **Metrics** - List of parameters with numeric values, which vary over time, for a particular scenario or simulation.
- **MVC** - Model view control framework.
- **CSV** - Coma separated values.