Testing Document

SWEN90082

Planimation

Team: PL-boxjelly

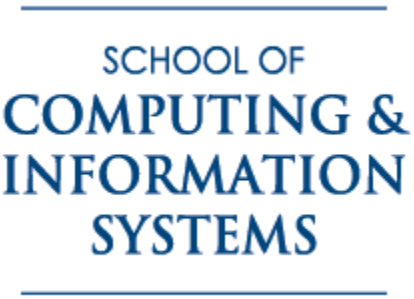
Bojing Zhou Bojing Zhou

Felipe Ramos Morales Felipe Ramos Morales

Shiqi Zhang Shiqi ZHANG

Xiaoyu Zhang XIAOYU ZHANG

Ziqi Meng Ziqi Meng



<students must use this table to track individuals' contributions to this document.

Every time you change this document, add the date you changed it, a description of your performed task and your name. For the version, please adopt the following format:

01.00-D<number> for draft versions related to Part 1 (any version before the final submission is considered draft). When your document is reviewed and finally ready to be submitted, change it to 01.00. For Part2, start with 02.00-D<number> and so on. This document should always be kept on GitHub>

Revision History

Date	Version	Description	Author
6/9/2021	01.00-D01	Initial draft, design functional test cases	Bojing Zhou
7/9/2021	01.00-D02	Functional test cases for US001	XIAOYU ZHANG
8/9/2021	01.00-D03	Functional test cases for US002	Felipe Ramos Morales
8/9/2021	01.00-D04	Functional test cases for US003	Shiqi ZHANG
9/9/2021	01.00-D05	Functional test cases for US006	Ziqi Meng
19/9/2021	01.00	Finish other test cases and entry data, final review and improvements on the document	Bojing Zhou

Contents

1. Introduction

1.1 Proposal

The purpose of this document is to define and present the test cases for project Platimation and team boxjelly, covering the test cases for the system use cases.

1.2 Target Users

This document in mainly designed for those responsible for executing the test cases in this project [team members and SWEN90082 teaching team].

1.3 Conventions, terms and abbreviations

This section explains the concept of some important terms that will be used throughout this document. These terms are described in the following table, presented in alphabetical order.

Term	Description

2. Covered Requirements

This section lists the system requirements covered in the test cases.

2.1 Functional or Product Requirements

Requirement Identifier (User Story ID)	Requirement Name
US001	As a user, I could access the main interface for access to four sub-modules (including generating the visualisation from problem files, generating the visualisation from VFG file, accessing the user manual and accessing the demo).
US002	As a user, I could upload domain, problem, and animation PDDL files for generating the visualisation of the plan (i.e. solution) of this planning problem.
US003	As a user, I could upload a VFG file for generating the visualisation directly.
US006	As a user, I could view the animation of the visualisation of a particular planning problem on the visualizer page after uploading the files.
US007	As a user, I could check each step of the plan, the status of any step in the animation by selecting a particular step, and the detailed step information of the selected step on the visualizer page.
US008	As a user, I could check the subgoals of each step and all the steps corresponding to a certain subgoal.
US009	As a user, I could view the visualization of the final goal state.
US010	As a user, I could check the visualization status of the previous or next step.

3. Functional Test Cases

This section describes the test cases that cover the product requirements of the system.

3.1 US001: Access the homepage

3.1.1 TC01-1: Jump to pages successful

Test Type:	**Execution Type:**
Functional	Manual
Objective: Verify if the links can route to certain pages successfully.	
Setup: · None	
Pre-Conditions: · None	

Notes: [1] Click the buttons to perform page jumps. [2] Use forward, backward to perform page jumps. [3] Refresh first then perform page jumps. [4] Use fast 3G mode to perform page jumps. [5] Use slow 3G mode to perform page jumps. * Corresponding pages loaded successfully
Time constraint: Minimum: <1s Maximum: ~10s

3.1.2 TC01-2: Jump to pages unsuccessful

Test Type: Functional	**Execution Type:** Manual
Objective: Verify that the links do NOT route to certain pages successfully.	
Setup: · None	
Pre-Conditions: · None	
Notes: [1] Try offline mode to perform page jumps. * A 404-error page should show if it's time out/the browser cannot load the page successfully	
Time constraint: Minimum: <1s Maximum: ~1min	

3.2 US002: Upload domain, problem, animation PDDL files

3.2.1 TC02-1: Upload domain file successful

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify if uploading a Domain file is performed successfully.	
Setup: · None	
Pre-Conditions: · None	

Notes: [1] Upload the Domain file. [2] Cancel the chosen file then upload again. [3] Upload the same Domain file after uploading one. [4] Upload another Domain file after uploading one. *The file should be uploaded successfully and the latter one should overwrite the former.
Time constraint: Minimum: <1min Maximum: 2 min

3.2.2 TC02-2: Upload domain file unsuccessful

Test Type:	**Execution Type:**
Functional	Manual/Automatic
Objective: Verify that uploading a Domain file is NOT performed successfully.	
Setup: · None	
Pre-Conditions: · None	
Notes: [1] Try to upload a file other than pddl extension. [2] Try to upload an empty pddl file. [3] Try to upload a pddl file with some language characters other than English. [4] Try to upload the file when it's offline. *The file should not be uploaded and there should be a warning for the user.	
Time constraint: Minimum: <1min Maximum: 2 min	

3.2.3 TC02-3: Upload problem file successful

Test Type:	**Execution Type:**
Functional	Manual/Automatic
Objective: Verify if uploading a Problem file is performed successfully.	
Setup: · None	

Pre-Conditions: · None
Notes: [1] Upload the Problem file. [2] Cancel the chosen file then upload again. [3] Upload the same Problem file after uploading one. [4] Upload another Problem file after uploading one. *The file should be uploaded successfully and the latter one should overwrite the former.
Time constraint: Minimum: <1min Maximum: 2 min

3.2.4 TC02-4: Upload problem file unsuccessful

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify that uploading a Problem file is NOT performed successfully.	
Setup: · None	
Pre-Conditions: · None	
Notes: [1] Try to upload a file other than pddl extension. [2] Try to upload an empty pddl file. [3] Try to upload a pddl file with some language characters other than English. [4] Try to upload the file when it's offline. *The file should not be uploaded and there should be a warning for the user.	
Time constraint: Minimum: <1min Maximum: 2 min	

3.2.5 TC02-5: Upload animation file successful

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify if uploading an Animation file is performed successfully.	
Setup: · None	

Pre-Conditions: · None
Notes: [1] Upload the Animation file. [2] Cancel the chosen file then upload again. [3] Upload the same Animation file after uploading one. [4] Upload another Animation file after uploading one. *The file should be uploaded successfully and the latter one should overwrite the former.
Time constraint: Minimum: <1min Maximum: 2 min

3.2.6 TC02-6: Upload animation file unsuccessful

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify that uploading an Animation file is NOT performed successfully.	
Setup: · None	
Pre-Conditions: · None	
Notes: [1] Try to upload an Animation file other than pddl extension. [2] Try to upload an empty pddl file. [3] Try to upload a pddl file with some language characters other than English. [4] Try to upload the file when it's offline. *The file should not be uploaded and there should be a warning for the user.	
Time constraint: Minimum: <1min Maximum: 2 min	

3.3 US003: Upload VFG file

3.3.1 TC03-1: Upload VFG file successful

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify if uploading a VFG file is performed successfully.	

Setup: · None
Pre-Conditions: · None
Notes: [1] Upload the VFG file. [2] Cancel the chosen file then upload again. [3] Upload the same file after uploading one. [4] Upload another file after uploading one. *The file should be uploaded successfully and the latter one should overwrite the former.
Time constraint: Minimum: <1min Maximum: 2 min

3.3.2 TC03-2: Upload VFG file unsuccessful

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify that uploading a VFG file is NOT performed successfully.	
Setup: · None	
Pre-Conditions: · None	
Notes: [1] Try to upload a VFG file other than vfg extension. [2] Try to upload an empty vfg file. [3] Try to upload a vfg file with some language characters other than English. [4] Try to upload the file when it's offline. *The file should not be uploaded and there should be a warning for the user.	
Time constraint: Minimum: <1min Maximum: 2 min	

3.4 US006: Visualise files

3.4.1 TC06-1: Parse VFG file successful

Test Type: Functional	**Execution Type:** Manual/Automatic

Objective: Verify if parsing the vfg file is performed successfully.
Setup: · VFG file has uploaded
Pre-Conditions: · None
Notes: [1] Parse the file *The file should be parsed successfully and the page jumps to the Demo page
Time constraint: Minimum: <1 min Maximum: 3 min

3.4.2 TC06-2: Parse VFG file unsuccessful

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify that parsing a VFG file is NOT performed successfully.	
Setup: · None	
Pre-Conditions: · None	
Notes: [1] Try to parse an empty vfg file. [2] Try to parse a vfg file with wrong format. [3] Try to parse a vfg file with correct format but empty values. [4] Try to parse the file when it's offline. *The system should pop up a warning about the error and stays in the current page.	
Time constraint: Minimum: <1min Maximum: 2 min	

4. Test Cases

This section describes the general test cases that will be related to more than one requirement or assumption / constraint, avoiding data replication.

4.1 TC07-1: Select plan steps

Test Type: Functional	**Execution Type:** Manual/Automatic

Objective: Verify if the displayed information is performed successfully when selecting.
Setup: · A correct animation pddl file is already loaded
Pre-Conditions: · None
Notes: [1] Select steps online. [2] Select steps offline. *The corresponding step information, subgoals should be displayed, and the animation should be played.
Time constraint: Minimum: <1s Maximum: 1s

4.2 TC08-1: Select subgoals

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify if the displayed information is performed successfully when selecting.	
Setup: · A correct animation pddl file is already loaded	
Pre-Conditions: · None	
Notes: [1] Select steps online. [2] Select steps offline. *The corresponding step information, subgoals should be displayed, and the animation should be played.	
Time constraint: Minimum: <1s Maximum: 1s	

4.3 TC10-1: Control animation

Test Type: Functional	**Execution Type:** Manual/Automatic
Objective: Verify if the displayed information is performed successfully when controlling.	
Setup: · A correct animation pddl file is already loaded	

Pre-Conditions: · None
Notes: [1] Play the animation from the start. [2] Select a step before playing the animation. [3] Select a subgoal before playing the animation [4] Select a step after playing the animation. [5] Select a subgoal after playing the animation [6] Select a step while playing the animation. [7] Select a subgoal while playing the animation [8] Play the animation after the end [9] Pause the animation [10] Select previous step [11] Select next step *The corresponding step information, subgoals should be displayed, and the animation should be played/paused.
Time constraint: Minimum: <1s Maximum: 1s

5. Entry Data

This section describes the entry data that will be used by more than one test case, avoiding data replication. These data are referenced by the test cases.

5.1 DATA VFG: <VFG Sample Data>

Description: The test vfg data format in json format.
{"visualStages": [**{** **"visualSprites": [** **{** **"prefabimage":"img-block",** **"showname": true,** **"x":300,** **"y":82,** **"color":{ "r":1.0, "g":0.98, "b":0.8, "a":1.0},** **"width":80,** **"height":80,** **"name":"b",** **"minX":0.286,** **"maxX":0.357,**

```
        "minY":0.091,
        "maxY":0.163
}
],
"stageName": "Initial Stage",
"stageInfo": "No Step Information"
"isFinal": "false"
}
  ],
  "subgoalPool": {
"m_keys": ["(on f g)", "(on c f)"],
"m_values": [
["f", "g"],
["c", "f"],
]
},
  "subgoalMap": {
"m_keys":[4, 5, 6],
"m_values":[
     [ "(on f g )" ], ["(on f g )" ],
]
  },
  "transferType":1,
  "imageTable": {
"m_keys":[
"img-claw",
    "img-block",
    "img-board"
],
"m_values": string[]
  },
  "message":""
}
```