**Hungry Student Test Cases – Milestone 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Priority** | **Expected Results** | **Actual Results\*** | **Pass / Fail** |
| 1.1  (Smoke  Test) | Check Map zoom  (reducing / incrementing). | 1. Go on selected Web Browser. 2. Reduce zoom of Map. 3. Increment zooms of Map. | Web Browser:  Google Chrome | Medium | Map API works correctly and is possible to zoom Map. | As expected | ✔ |
| 1.2  (Smoke  Test) | Check Position on Map. | 1. Check, by clicking on Position, if it shows correct device IP Position. | Web Browser:  Google Chrome | Medium | Map API works correctly, especially device IP Position. | As expected | ✔ |
| 1.3  (Smoke Test) | Check Map visualisation. | 1. Check if Map is shown on listed Web Browsers. | Web Browser:  Google Chrome | High | Map API works correctly for visualisation on Web Browser. | As expected | ✔ |

**Hungry Student Test Cases – Milestone 2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Priority** | **Expected Results** | **Actual Results\*** | **Pass / Fail** |
| 2.1  (Integration  Test) | Check Database connection. | 1. Set in JUnit Tests SQL queries. 2. Check if connection to Database works. | JUnit 4 on Eclipse/IntelliJ IDEA &  MySQL Workbench | Medium | JDBC Class & MySQL Database works properly. | As expected | ✔ |
| 2.2  (Unit Test) | Check “Filter” criteria. | 1. Set in JUnit Tests “Vegetarisch”, “TakeAway”and both Filter. 2. Set in JUnit Tests “Preis” Filter. Combine with listed in step 1. 3. Set in JUnit Tests “Küche” (differents). Combine with listed in step 1. & 2. | Eclipse/IntelliJ IDEA  Test Tool: JUnit 4 & Selenium | Low | “Filter” criteria work. | As expected | ✔ |
| 2.3  (Unit Test) | Check “Markierungen”, according to CRUD procedure (Create Read Update Delete). | 1. Create a “Markierung”. 2. See this “Markierung” if is in Database. 3. Update this “Markierung” with new data. 4. Finally delete created “Markierung”. | Eclipse/IntelliJ IDEA  Test Tool: JUnit 4 & Selenium | High | “Markierung” work. | As expected | ✔ |
| 2.4  (Unit Test) | Check implemented methods in all Java classes. | 1. Set in JUnit Tests, by testing all listed methods in all Java classes. 2. Check if works (test pass). | JUnit 4 on Eclipse/IntelliJ IDEA | High | Implemented methods in all Java classes works. | As expected | ✔ |

**Hungry Student Test Cases – Milestone 3**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Priority** | **Expected Results** | | **Actual Results\*** | | **Pass / Fail** | |
| 3.1  (Unit Test) | Check “User”, according to CRUD procedure  (Create Read Update Delete). | 1. Create a “User”. 2. See this “User” if is in Database. 3. Update this “User” with new data. 4. Finally delete created “User”. | Eclipse/IntelliJ IDEA  Test Tool: JUnit 4 & Selenium | High | “User” work. | | As expected | | ✔ | |
| 3.2  (Unit Test) | Check “Favoriten”, according to CRUD procedure  (Create Read Update Delete). | 1. Create a “Favorit”. 2. See this “Favorit” if is in Database. 3. Update this “Favorit” with new data. 4. Finally delete created “Favorit”. | Eclipse/IntelliJ IDEA  Test Tool: JUnit 4 & Selenium | High | “Favoriten” works. | | As expected | | ✔ | |
| 3.3  (Smoke Test) | Check Web UI main functions (User, Login, Favoriten & Markierungen). | 1. Try to create a User, to Login and Logout. 2. Relogin with current created user and try to create a “Markierung”, set this “Markierung” to “Favoriten”. Finally deleted them from “Favoriten” and from “Erstellte Markierungen”. | Web Browser:  Google Chrome  Test Tool: Selenium | Low | Web UI work. | | As expected | | ✔ | |
| 3.4  (System Test) | Check implemented methods in all Java classes. | 1. Re-run all Unit & Selenium Tests. 2. Check if works (test pass). | Eclipse/IntelliJ IDEA  Test Tool: JUnit 4 & Selenium | High | New features work with implemented methods in Java classes. | | As expected | | | ✔ |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Priority** | | **Expected Results** | | **Actual Results\*** | **Pass / Fail** | |
| 3.5  (Smoke Test) | Check “Login” with valid User. | 1. Create a User. 2. Login with this created User. Try as first with a wrong password. See error message. Then login with correct Password. 3. Logout. 4. Check if User is listed in Database. 5. Set a new Password for current user. 6. Logout and re-login to check if works. Finally, Logout. | Web Browser:  Google Chrome  Test Tool: Selenium | Medium | | Valid “Login” works  (positive test). | | As expected | ✔ | |
| 3.6  (Smoke Test) | Check “Login” with invalid User. | 1. Login with an invalid User (not created user or. not listed in Database). | Web Browser:  Google Chrome  Test Tool: Selenium | Medium | | Invalid “User” will not work  (negative test). | | As expected | ✔ | |

**Hungry Student Test Cases – Milestone 4**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Priority** | **Expected Results** | **Actual Results\*** | **Pass / Fail** |
| 4.1  (Unit Test) | Check “Rezensionen”, according to CRUD procedure  (Create Read Update Delete). | 1. Create a “Rezension”. 2. See this “Rezension” if is in database. 3. Update this “Rezension” with new data. 4. Finally delete created “Rezension”. | Eclipse/IntelliJ IDEA  Test Tool: JUnit 4 & Selenium | Medium | “Rezensionen” works. | As expected | ✔ |
| 4.2  (Unit Test) | Check “Erweiterte Registration” functions. | 1. Use same Tests of “User” (See MS3: 3.1), by adding new implemented features on “Registration”. | Eclipse/IntelliJ IDEA  Test Tool: JUnit 4 & Selenium | Medium | “Erweiterte Registration” works with already implemented User “Registration”. | Not supported!   * No Test! | NA |
| 4.3  (Integration Test) | Check Mobile compatibility. | 1. Try to do some Web UI functions in Mobile device. 2. Check if it works. | Android Device emulator (Android Studio) | Low | “Mobile” or. Android compatibility works. | Not supported!   * No Test! | NA |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Priority** | **Expected Results** | **Actual Results\*** | **Pass / Fail** |
| 4.4  (Acceptance Test or. System Test) | SUT (System Under Test):  Check if developed software accords to Acceptance criteria of Technical documentation. | 1. Check if developed software respect “Acceptance criteria” (explorative test) of Technical documentation (see “Pflichtenheft”). | Eclipse/IntelliJ IDEA  Web Browser:  Google Chrome  Test Tool: JUnit 4 & Selenium | High | Software respect “Acceptance criteria”. | As expected | ✔ |
| 4.5  (Regression Test) | REG (Regression Test):  Recheck implemented features. | 1. Run Unit Tests with JUnit. 2. Run Selenium Tests. 3. Explorative Test. | Eclipse/IntelliJ IDEA  Web Browser:  Google Chrome  Test Tool: JUnit 4 & Selenium | High | Complete Software works. | As expected | ✔ |
| 4.6  (Regression Test) | REG (Regression Test):  Recheck corrected issues. | 1. Recheck if works corrected issues. 2. Run Unit Tests with JUnit. 3. Run Selenium Tests. 4. Do short Explorative Test. | Eclipse/IntelliJ IDEA  Web Browser:  Google Chrome  Test Tool: JUnit 4 & Selenium | High | Complete Software works (with corrected issues, if available). | As expected | ✔ |
| 4.7  (Smoke Test) | Check final software working on Tomcat Server 8.5.35. | 1. Open Eclipse / IntelliJ IDEA. 2. Start Tomcat Server. 3. Open Web Browser. 4. Check final Software before Release. | Eclipse/IntelliJ IDEA, Web Browser & Tomcat Server | High | Hungry Student App works! | As expected | ✔ |