Test Plan

# **Test Requirements**

* Treating the functional and nonfunctional requirement specified in the document ***KCPM-SuperMarket-SRSv1.1.pdf*** as tight user requirements to follow.
* Organize test suites according to these user requirements.
* Techniques to use:

1. Manual testing
2. Link Checking using <https://www.drlinkcheck.com/>
3. Compatibility Checking using <https://www.powermapper.com/products/sortsite/checks/browser-compatibility/>
4. Automation testing using Selenium IDE
5. Performance testing using JMeter

# **2. Test Input-Output**

* Common input: The app

1. **Manual Testing**

* Input:
* SQL Database
* List of test suites, test cases, bugs and test report sheets to fill
* Output: Fill the input test documents

1. **Link Checking**

* Input: <https://www.drlinkcheck.com/>
* Output:
* Link Checking report (view on the tool page)
* Broken Link report

1. **Compatibility Checking**

* Input: <https://www.powermapper.com/products/sortsite/checks/browser-compatibility/>
* Output:
* Compatibility Checking report

1. **Automation Testing**

* Input: Selenium IDE
* Output: Saved test project file for each tested function

1. **Performance Testing**

* Input: JMeter
* Output:
* Saved test plan file for each tested function
* Summary report with metrics for each tested function, along with test script

# **3. Testing Environment**

1. The app Homepage URL: <http://mymarket-2.apphb.com/>
2. AppHarbour project where the app is deployed:

<https://appharbor.com/applications/mymarket-2>

1. Connection info of SQL database in use from AppHarbour:

<https://sequelizer.apphb.com/database>

1. The app release repository with reference documents:

<https://drive.google.com/drive/folders/1sCbIYD-ZNn9rrq4ZmGpwtkSTcnA__ipU?usp=sharing>

1. The Test Project repository where all Testing documents and artefacts are stored:

<https://drive.google.com/drive/folders/11ZvcW3KxzdvPB-YhxCzbpFNeM7AYLPIL?usp=sharing>

1. The testing tools mentioned above and their corresponding environment

# **4. Dividing workload and Schedule**

* Most items are the score sheets are evenly divided between 2 team members
* Estimated schedule:

1. Deploy the app (1st day)
2. Prepare the testing document form and template (2nd day)
3. Learn the various tools basics (2nd day)
4. Manual Testing (3rd and 4th day)
5. Link Checking (5th day)
6. Compatibility Checking (5th day)
7. Automation testing (5th day)
8. Performance Testing (5th day)

# **5. Risk management**

1. **Risk factor #1: Only black-box testing is performed**

* Testing is done without web server source code (compiled to binary beforehand).
* Therefore, some aspects of white-box testing are excluded.
* Particularly, internal data structure and logic are not tested and may have bugs.
* The proposed solution is to also perform Unit Testing and white-box Integration testing.

1. **Risk factor #2: Time constraint, not 100% complete testing**

* Due to time constraint, members’ workload is not completed 100%.
* Therefore, testing items may not be thoroughly checked.
* As a result, a few bugs may stay uncovered.
* The solution is to maintain a good schedule.

1. **Risk factor #3: Mismatch in installation and deployment**

* Reference documents are not specific enough, so mismatch in installation and deployment of the app may occur.
* As a result, a different version of the app than the original intended one may be tested.
* As a result, the testing outputs may not correctly reflect the true app status.
* Solution: communicate sufficiently with the team that handed down the reference documents and app release.

1. **Risk factor #4: Mismatch between Customer Requirement and intended Implementation**

* Mismatch between app implementation and customer requirements may be founded.
* The testing team may not know if certain behaviour of the app which conflicted with the requirements is intended or not, to be implemented if missing or not, and may treat these as bugs.
* As a result some bugs recorded by the testing team may be waivable, or not reflect a real bug.
* Solution: communicate sufficiently with the upstream team.