<GAP Hardware online project>

SYSTEM REQUIREMENTS SPECIFICATIONS

Version <1.0>  
<30/08/2018>

Shalini Raigir

Swapna Jaidi

Sowmya Vasantha Rao

Srivani Nampally

[1. Introduction 3](#_Toc512869777)

[1.1 Purpose 3](#_Toc512869778)

[1.2 Product Scope 3](#_Toc512869779)

[1.3 References 3](#_Toc512869780)

[2. Overall description 3](#_Toc512869781)

[2.1 Product Functions 3](#_Toc512869782)

[2.2 Operating Environment 3](#_Toc512869783)

[2.3 Design and Implementation Constraints 3](#_Toc512869784)

[2.4 User Documentation 4](#_Toc512869785)

[2.5 Security Requirements 4](#_Toc512869786)

[2.6 Operating Environment 4](#_Toc512869787)

[3. System features 4](#_Toc512869788)

[3.1 Product Functions 4](#_Toc512869789)

[3.2 Logical Database Requirements 5](#_Toc512869790)

[3.3 Server type requirements 5](#_Toc512869791)

[4. Functional requirements 5](#_Toc512869792)

[5. Non-functional Requirements 6](#_Toc512869793)

[5.1 Performance Requirements 7](#_Toc512869794)

[5.2 Safety Requirements 7](#_Toc512869795)

[5.3 Security Requirements 7](#_Toc512869796)

[5.4 Software Quality Attributes 7](#_Toc512869797)

[5.5 Business Rules 7](#_Toc512869798)

[6. Other Requirements 7](#_Toc512869799)

[7. Appendix A: Glossary 8](#_Toc512869800)

[8. Appendix B: Analysis Models 8](#_Toc512869801)

# Introduction

## Purpose

Identify the project whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.

GAP Harware webpage will act as an interface to the All the Clients of GAP Harware as well as to the Public.

Purpose of this webpage is to create a multiple page website especially for the Wholesale Hardware Distribution. Public with valid credentials could check the avaible product information and prices.

It’s main purpose is to make hardware product information and related forms available to the GAP Harware clinets for the wholesale distribution and will be for ease of use for their clients, for their convenience.

## 1.2 Product Scope

Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.

Please Refer section 1.1, section 2, section 3 , section 4 and section 5 for the requirements

## 1.3 References

List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

Please Refer section 1.1

# Overall description

## 2.1 Product Functions

Summarise the major functions the product must perform or must let the user perform. Details will be provided in the following sections, only a high level summary (such as a bullet list) is needed here. Organise the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram.

This Website must enable established clients or user to login to the GAP Harware webpage.

This website would enable Users to check Product Information upon successful login.

It also enables users to download order forms, Application forms or any other forms.

It would enable users to check contact details of the GAP Hardware.

It would enable users to logout of the system

## 2.2 Operating Environment

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

Windows operating system would be the operating environment.

## 2.3 Design and Implementation Constraints

Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).

We would use various tools to develop the webpage and yet to be decided

We would use MySQL as a database.

We would use PHP, CSS and HTML for creating webpage.

We Would create wireframe diagrams using word.

We would use a XAMPP apache server

Developer will responsible for devloping the software.

Testers will be responsible for testing the software.

Client organization will be responsible for maintaing thr delivered software.

## 2.4 User Documentation

List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

User documentation will be completed while devloping the webpage.

## 2.5 Security Requirements

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

Because of the security reasons Gap Hardware will create password and email to the clients when they forget.

## 2.6 Operating Environment

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

Windows would be the Operating environment

# System features

This section illustrates organising the functional requirements for the product by system features, the major services provided by the system. You may prefer to organise this section by use case or user story.

## 3.1 Product Functions

Provide a summary of the major functions that the product will perform. Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product.

For clarity:

* The functions should be organised in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.
* Textual or graphic methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product but simply shows the logical relationships among variables.

This describes the functionality of the system in the language of the customer. What specifically does the system that will be designed have to do? Drawings are good, but remember this is a description of what the system needs to do, not how you are going to build it. (That comes in the design document).

Please refer section 2.1

## 3.2 Logical Database Requirements

This section specifies the logical requirements for any information that is to be placed into a database. This may include:

1. Types of information used by various functions
2. Frequency of use
3. Accessing capabilities
4. Data entities and their relationships
5. Integrity constraints
6. Data retention requirements

If the customer provided you with data models, those can be presented here. ER diagrams (or static class diagrams) can be useful here to show complex data relationships. Remember a diagram is worth a thousand words of confusing text.

GAP harware login and password information need to be stored in the database for processing.

## 3.3 Server type requirements

This section is for any information required relating to the hosting of the site and the server types required. Identify and outline: application, backup, email, firewall, proxy, web

* Web page would need webserver and host.

# Functional requirements

This section contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system and all functions performed by the system in response to an input or in support of an output. The following principles apply:

Specific requirements should be stated with all the characteristics of a good SRS

* Correct: All the information which would display on the website need to be correct.
* Consistency: Header, Logo and footer of the website need to be Consistent throughout all the pages.
* Modifiable: Web site need to be developed in such a way that, it could be easily modifiable as per the client needs.
* Complete:Website need to be devloped completely to accommodate or accomplish client requirements.
* Availability: Web page needed to be available and accessible to the GAP Hardware users/Clients, public and staff all the time.

Specific requirements should be cross-referenced to earlier documents that relate

All requirements should be uniquely identifiable (usually via numbering like 3.1.2.3)

Careful attention should be given to organising the requirements to maximize readability

Before examining specific ways of organising the requirements it is helpful to understand the various items that comprise requirements as described in the following subclasses. This section reiterates previous sections, but is for developers not the customer.

**Remember this is not design**. Do not require specific software packages, etc unless the customer specifically requires them. Avoid over-constraining your design.

Use proper terminology:

* The system shall… A required, must have feature
* The user shall…..

**Each requirement should** be uniquely identified for traceability. Usually, they are numbered 3.1, 3.1.1, 3.1.2.1 etc. Each requirement should also be testable. Avoid imprecise statements like, “The system shall be easy to use” & “The system shall be developed using good software engineering practice”

**Avoid examples,** This is a specification, a designer should be able to read this spec and build the system without bothering the customer again. Don’t say things like, “The system shall accept configuration information such as name and address.” The designer doesn’t know if that is the only two data elements or if there are 200. List every piece of information that is required so the designers can build the right UI and data tables.

Use the table below:

**Requirement #– Login Screen**

|  |  |
| --- | --- |
| Description | The user shall be able to …..  The system shall be able to….. |
|  |  |
|  |  |
| Input | User should be able to login with valid credentials. |
| Processing | System should be able to validate login and password |
| Output | Successful login navigates to the GAP Hardware website |
|  |  |

## Products page

|  |  |
| --- | --- |
| Description |  |
| Input | Authorised user should be able to click on products page |
| Processing | System should be able to open products page |
| Output | User should be able to check products |
|  |  |
|  |  |

**ABOUT US PAGE**

|  |  |
| --- | --- |
| Description |  |
| Input | User should be able to click on about us page |
| Processing | System should be able to open about us page |
| Output | User should be able to open about us page |
|  |  |
|  |  |

**CONTACT US**

|  |  |
| --- | --- |
| Description |  |
| Input | User should be able to click on contact us page |
| Processing | System should be able to open contact us page |
| Output | User should be able to open contact us page |
|  |  |
|  |  |

**LOGOUT**

|  |  |
| --- | --- |
| Description |  |
| Input | User should be able to click on logout link |
| Processing | System should be able to forward to logout page |
| Output | User should be able to see logout page |
|  |  |
|  |  |

# Non-functional Requirements

If there are non-functional requirements for the system under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. For example: Product properties, constraints on the services or functions, property or quality the system must have and or any attributes of the system.

Accessability: Documents and other forms will be easily accessable by employees

Testability:During developing the webpage , it is easily testable.

Reusablity: After this web page is developed client might reuse the web page with other web pages of GAP Hardware depending on their need.

## 5.1 Performance Requirements

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

Performance:Web would loads with out any delays however document or form size limits it’s performance.

## 5.2 Safety Requirements

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.

During the development care must be taken from damage or loss of information provided by client.

There is a need to have a back up while developing webpage to avoid loss of user information and other files

## 5.3 Security Requirements

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

In an important form of risk management, you need to prioritize security requirements to determine the appropriate level of security. Project teams should evaluate the likelihood of security threats (such as unauthorized access to data or network resources), identify the appropriate security controls to mitigate the risk, and test early and often to ensure that the controls are established.

The User identity authentication requirements for GAP Hardware online store are LOGIN credentials.

These are unique for each customer.

Some security requirements related to sensitive software development procedures, such as preventing fraud or harm when code is migrated into production. Separation of duties would be an appropriate security control for these requirements; it means that a person who develops a piece of code cannot also move that piece of code into the production environment.

The product should satisfy the following security and privacy certifications:

## Certified Information Systems Security Professional (CISSP)

Systems Security Certified Practitioner (SSCP)

## 5.4 Software Quality Attributes

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

Flexibility: Web page to be created will be flexible to the Gap hardware users.

Usability: It is needed to be created for ease of use and convenience

Testability: Web page will be testable once it developed.

Reusability: Once web page is developed, it can be reusable by the Client in the future.

Maintainability: This web page would be maintained by Client, once developed.

Reliability – the website needs to be reliable in such that it needes to be loaded correctly.

## 5.5 Business Rules

List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.

GAP Hardware client logins would be unique to each individual

Only established clients must be able to check product information and prices.

All forgotten password requests need to be requested through email to admin.

There will be only limited no.of products available in the products page.

Only established can able to download the order form.

Only established clients can order the products using order form.

# Other Requirements

Define any other requirements not covered elsewhere in the SRS. This might include internationalisation requirements, legal requirements, reuse objectives for the project, and so on.

Add any new sections that are pertinent to the project.

All agreed requirements are gathered and scoped into the project, any further requirements will be treated as a change request and will be dealt at the later stage of the project.

# Appendix A: Glossary

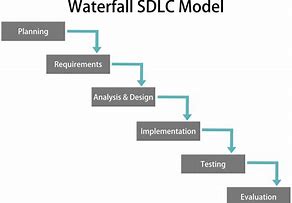
Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.

|  |  |
| --- | --- |
| Term | Meaning |
| ​Approval | ​Formal, recorded acceptance of one or more outputs/recommendations/documents by an authorised Person.  Programme/project documents e.g. Project Management Plan (PMP) need to be formally approved as described in respective frameworks. |
| Assurance | ​Programme or project control(s) that deal with known issues. Assurance activities and processes aim to ensure that an outcome or product will be completed in spite of expected or known issues. |
| ​Budget | ​ Funds allocated to a programme/project or area. |
| Business Owner  Also see: Product Owner | ​Receiver of a project’s deliverables including products, documentation, handover artefacts, operational risks etc. |
| ​Business Requirement(s) | ​Formalised project need(s) that have been approved by Business Owners and SROs, and agreed to by ICT teams. |
| Closing Project (status reporting) | ​A departmental reference to projects handing over products, documentation and outstanding risks/issues.  From a project status reporting perspective, the term signifies SRO approval for a project’s close phase, as well as submission of its last monthly status report. |
| Costing | ​Calculating funds needed for an initiative.  Costing is a primary component of business planning, and all programmes/projects go through one or more costings. |
| ​Closure Report | Report submitted in a project’s Close Phase |
| ​Delivery | ​Handing over project outputs/products/services to a Business Owner or an ICT release.  In ICT environments this may mean code deployment. In other environments, this may be the point of handing over all outputs, products and/or services to the Business Owner. |
| Deliverable | ​An output/product of a project.  Deliverables may include all formal project management documentation, handover documents, final products/services produced by the project etc. The usual recipients of deliverables include project sponsors, records management areas and Business Owners. |
| Hypothesis (pl. hypotheses) | ​An abstract or theoretical concept that is ready to be tested i.e. a concept model. Once tested, the hypothesis is considered to have either been proven true, or false.  Hypothetical models are usually the starting point for prototype development. |
| ​Impact | ​The effect of an activity, decision or event.  Impacts are assessed for project elements such as risk management, stakeholder engagement etc. |
| Issue | ​When a risk eventuates into a situation and its consequences are realised.  Usually, issue management flows on from risk management, and in many cases is a component of risk management |
| Key Milestone  Also see: Milestone | ​Project life cycle event of critical interest to stakeholders.  Many management reports also feature key milestones. Programme, project and BAU schedules feature milestones that depict achievements, key dates or phases in business cycles or product/services development. The difference between a milestone and a key milestone in a schedule is the element of stakeholder interest in a key milestone. |
| Methodology | ​Collection of methods that complement each other towards a common objective.  In common practice, methodologies are created when methods from different disciplines e.g. software engineering methods and team management methods, are brought together for a purpose such as managing a large project with multiple ICT components |
| Milestone  Also see: Key Milestone | ​Programme/project life cycle event of interest to stakeholders.  Some milestones may be major task completion dates, others may be multilateral agreements between stakeholders. What sets a milestone apart from other events is its relevance to project interests. |
| Outcome | ​The results, impacts or consequences of a purpose or activity, as defined in the annual Appropriation Acts and the portfolio budget statements, by a Commonwealth entity and company.  In programme and project environments, outcomes are the combined effect of one or more outputs, products or services on an entity e.g. the department. |
| ​Planning Phase | ​The second phase in a departmental project’s life cycle. |
| ​Project | ​A set of time boxed activities producing a unique output. |
| ​Quality | ​The suitability of something for intended purpose(s).  In P3M environments, quality could relate to functionality, usability, low maintenance costs or a multitude of other factors. Quality Framework is meant to assist business areas to manage the quality of their products, services and processes from design to implementation. |
| Resource | ​Staff, financial or other asset.  In financial contexts and some P3M environment, various components, roles and people are represented as resources for planning purposes. |
| Risk | ​The effect of uncertainty on objectives.  Uncertain positive/negative events or factors may have an effect on objectives that may often jeopardise the course of a programme/project, and the department’s Risk Management Policy explains how to deal with risks and manage them. |
| Stakeholder | ​An individual or entity who may have an interest or influence the department’s service delivery arrangements, programmes or projects.  These include our customers, staff, Commonwealth and State/Territory governments, other government agencies, community organisations/third sector organisations, professional organisations, non-government businesses and unions. |
| Stand Up | ​A daily meeting held at the same time that brings everyone up to date on current work. |
| User Acceptance Testing (UAT) | ​Final pre-release software testing by end users. |
| Work Breakdown Structure | ​A hierarchical map of a project schedule.  Project schedules show resource allocations, across the life cycle of that project, the tasks that these resources will complete, the service/product components that will be output by these tasks and so on. A WBS structures and illustrates this information logically to facilitate planning and scheduling. |

# Appendix B: Analysis Models

Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.*

After the design draft approval this section will be updated with latest data flow diagrams, entity relationship diagrams, use cases and screenshots etc.

[](https://www.bing.com/images/search?view=detailV2&ccid=IZy3s/Q%2b&id=E8AA6FB9F7F72391AFB10653973E48F8065B817A&thid=OIP.IZy3s_Q-naj5P1oWtp_xUAHaFa&mediaurl=https://oli.cmu.edu/repository/webcontent/ac80970d80020ca60078c7f8185aefe0/_u5_hims_implementation_management/_m1_systems_implementation/webcontent/u5_m1_pg4b/image1.jpeg&exph=684&expw=935&q=waterfall+methodology&simid=608023838139810058&selectedIndex=30)

