**Software Requirements Specification**

**for**

Life Habitat

**Version 1.0 approved**

**Prepared by**

**PM:** Brianna Gannett

**Designer:** RoisinRumsey

**Developers:** Sarah Turmel

Dean Hauser

Emily Scott

**Team Ethernet**

**10/7/2024**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Sarah Turmel | 10/8/24 | Filling in features and descriptions of features | 1.1 |
| Emily Scott | 10/9/24 | Filled in Main Page | 1.2 |
| Roisin Rumsey | 10/9/24 | Brought ½ Functional Reqs. from reqs. Doc. | 1.3 |
| Dean Hauser | 10/9/24 | Brought ½ Functional Reqs. from reqs. Doc. | 1.3 |
| Sarah Turmel | 10/9/24 | Brought ½ NonFunc. Reqs. from reqs. Doc | 1.4 |
| Brianna Gannett | 10/9/24 | Brought ½ NonFunc. Reqs. from reqs. Doc | 1.4 |
| Brianna Gannett | 10/9/24 | Added UI Mockups To Section 3.1 | 1.5 |

# **Introduction**

## **Purpose**

*<Identify the product 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.>*

## **Document Conventions**

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## **Intended Audience and Reading Suggestions**

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## **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.>*

## **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.>*

# **Overall Description**

## **Product Perspective**

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## **Product Functions**

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize 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 or object class diagram, is often effective.>*

## **User Classes and Characteristics**

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## **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.>*

## **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).>*

## **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.>*

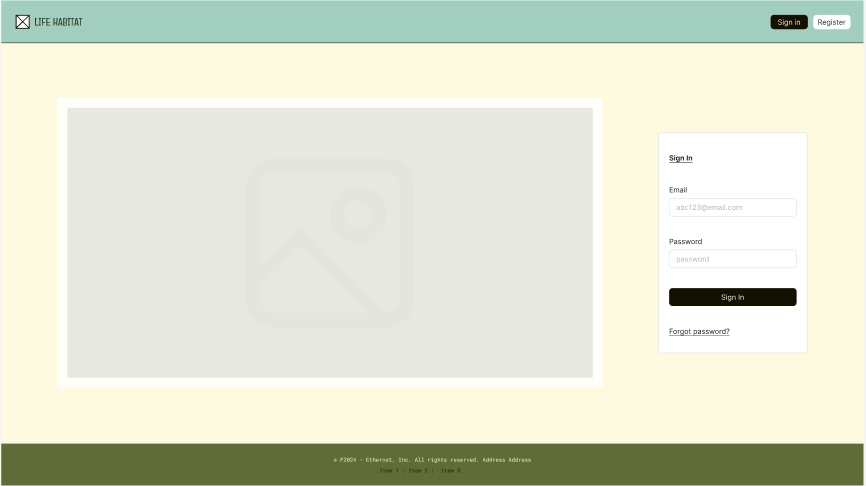
## **Assumptions and Dependencies**

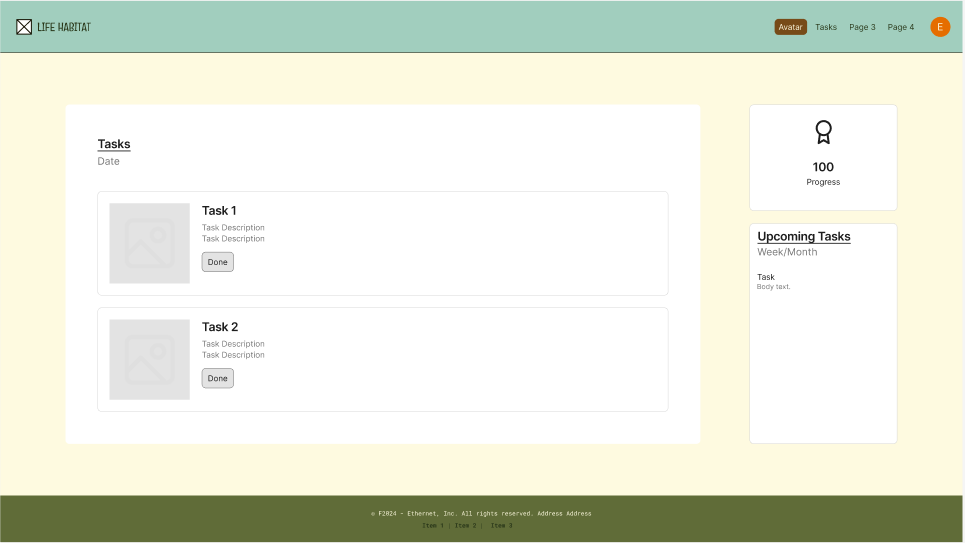
*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# **External Interface Requirements**

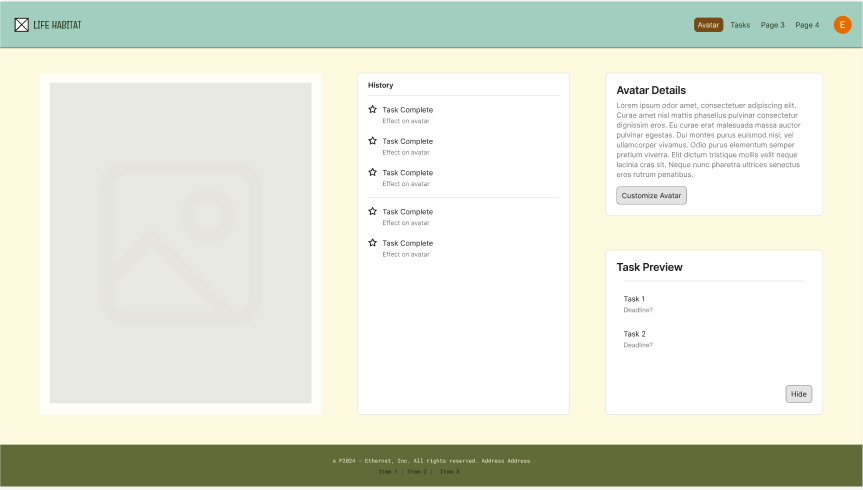
## **User Interfaces**

This is an example of the Life Habitat Sign In page. Clicking the Register button at the top of the page will allow the user to create an account if they do not already have one. Clicking Forgot Password will allow the user to send an email to the email address associated with their account to create a new password.



This is an example of Life Habitat’s Task Page. From this page you can view your current open tasks and see upcoming tasks on the right. You can mark tasks as completed by pressing the ‘Done’ button under their title and description.

This is an example of the avatar Page. On this page you can see the avatar (where the gray box currently appears on the left) and its details. You can also preview upcoming tasks and see the history of tasks you have completed.



## **Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## **Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## **Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# **System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## Tasks

4.1.1 Description and Priority

This feature will allow the user to create and modify tasks with different characteristics and to satisfy different needs. Users will be able to specify what a task is for, whether or not it repeats, and more. This feature is High Priority.

4.1.2 Stimulus/Response Sequences

*<NOT FOR DELIVERABLE 1>*

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.1.3 Functional Requirements

FREQ-1.1: The system shall track user-created tasks.

FREQ-1.2: The system shall provide a space for the user to enter a description of each task.

FREQ-1.3: The system shall repeat daily or weekly tasks accordingly.

FREQ-1.4: The system shall allow users to get task recommendations through AI generation.

FREQ-1.5: The system shall request to send the user push notifications for task reminders

FREQ-1.6: The system shall allow users to edit set tasks as needed.

FREQ-1.7: The system shall recommend new healthy habits to the task list once a user has successfully completed previous repeated tasks regularly

## User Encouragement

4.2.1 Description and Priority

This feature will provide the user with encouragement to complete or after completion of their tasks and a friendly atmosphere. Users will receive notifications as often as specified to complete a task and positive feedback on their progress. This feature is High Priority.

4.2.2 Stimulus/Response Sequences

*<NOT FOR DELIVERABLE 1>*

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.2.3 Functional Requirements

FREQ-2.1: The system shall have the avatar provide additional encouragement when the user fails to complete a task.

FREQ-2.2: The system shall provide encouraging feedback to the user when a task is failed.

FREQ-2.3: The system shall reward users for completing tasks without the use of reminders by providing achievements or points when a task is marked completed by the user.

FREQ-2.4: The system shall motivate the user to build positive habits with a customizable avatar that reflects the user’s progress and habits.

## Avatar

4.3.1 Description and Priority

The application will include a customizable avatar to reflect the user’s habits in caring for themself. Failing to complete tasks will lead to a more disgruntled appearance, whereas being timely and making progress will lead to an improvement in the avatar’s health. This feature is High Priority.

4.3.2 Stimulus/Response Sequences

*<NOT FOR DELIVERABLE 1>*

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.3.3 Functional Requirements

FREQ-3.1: The system shall allow users to customize their avatars with any unlocked rewards.

FREQ-3.2: The system shall motivate the user to build positive habits with a customizable avatar that reflects the user’s progress and habits.

FREQ-3.3: The system shall allow the user to buy avatar skins with their achievement points.

## Account

4.4.1 Description and Priority

The application will require that users create accounts. This will allow for the system to track user progress and save user data. This feature is High Priority.

4.4.2 Stimulus/Response Sequences

*<NOT FOR DELIVERABLE 1>*

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.4.3 Functional Requirements

FREQ-4.1: The system shall allow users to sign up with an email and password.

FREQ-4.2: The system shall allow users to edit their profile to add personal information.

FREQ-4.3: The system shall allow users to delete their accounts.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

NFREQ 1.1 The system shall allow for up to 100,000 simultaneous users while maintaining optimal performance.

NFREQ 1.2 The system shall allow for quick and easy navigation to its most meaningful features from any page.

NFREQ 1.3 The system shall have a page load time of under 3 seconds in 95% of all instances within any given month.

NFREQ 1.4 The system shall run on major web browsers like Chrome, Safari, and Firefox.

NFREQ 1.5 The system shall ensure consistent performance across different screen sizes to provide a seamless user experience.

## **Safety Requirements**

*There are currently no safety requirements associated with Life Habitat*

## **Security Requirements**

NFREQ 3.1 The system shall encrypt user data to protect it from other users and third parties.

NFREQ 3.2 The system shall follow the General Data Protection Regulation (GDPR).

NFREQ 3.3 The system shall check for user idle time and automatically log out after an hour of inactivity

## **Software Quality Attributes**

NFREQ 4.1 The system shall be able to send or receive data from servers within 30 seconds 99% of the time.

NFREQ 4.2 The system shall be online 95% of the time with the exception of scheduled maintenance blocks.

NFREQ 4.3 The system shall delete accounts that have been dormant for more than two years.

NFREQ 4.4 The system shall recover from any errors quickly without major data loss or user interruptions.

NFREQ 4.5 The system shall backup every hour to make sure no data is lost

## **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.>*

# **Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to 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.>*

**Appendix B: Analysis Models**

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

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*