**PhiLance**

**SYSTEM FUNCTIONAL SPECIFICATION**

**Table of Contents**

[1 PhiLance 4](#_Toc517714379)

[1.1 Function Data 4](#_Toc517714380)

[1.1.1 Function Group 4](#_Toc517714381)

[1.1.2 Function Owner 4](#_Toc517714382)

[1.2 Revision notes 4](#_Toc517714383)

[1.3 Introduction 4](#_Toc517714384)

[1.3.1 Scope 4](#_Toc517714385)

[1.3.2 Not in Scope 4](#_Toc517714386)

[1.3.3 General Description 4](#_Toc517714387)

[1.3.4 Guiding Principles 5](#_Toc517714388)

[1.4 I/O 5](#_Toc517714389)

[1.5 Information 5](#_Toc517714390)

[1.6 Functional Requirements 5](#_Toc517714391)

[1.6.1 Use Cases 5](#_Toc517714392)

[1.6.1.1 List of PhiLance Roles/Actors 5](#_Toc517714393)

[1.6.2 PhiLance 6](#_Toc517714394)

[1.6.2.1 Use Cases 6](#_Toc517714395)

[1.6.2.1.1 Use Case: Create Account 6](#_Toc517714396)

[1.6.2.1.2 Use Case: Login 8](#_Toc517714397)

[1.6.2.1.3 Use Case: Browse Project 9](#_Toc517714398)

[1.6.2.1.4 Use Case: Browse Project 10](#_Toc517714399)

[1.7 Architecture and Flows 12](#_Toc517714400)

[1.7.1 System Architecture Diagram 12](#_Toc517714401)

[1.7.2 Description of Interfacing Systems 12](#_Toc517714402)

[1.7.2.1 Future System Integration 12](#_Toc517714403)

[1.8 Other Requirements 12](#_Toc517714404)

[1.8.1 Operational 12](#_Toc517714405)

[1.8.2 Legal Considerations 12](#_Toc517714406)

[1.8.3 Tax Considerations 12](#_Toc517714407)

[1.9 Performance Specifications 12](#_Toc517714408)

[1.10 Security and Integrity Requirements 12](#_Toc517714409)

[1.10.1 Infrastructure Security 12](#_Toc517714410)

[1.10.2 Product Security 12](#_Toc517714411)

[1.10.3 Corporate & Personnel Security 12](#_Toc517714412)

[1.10.4 Compliance & Privacy 13](#_Toc517714413)

[1.11 Appendix 13](#_Toc517714414)

[1.11.1 Acronyms 13](#_Toc517714415)

# PhiLance

## Function Data

### Function Group

### Function Owner

## Revision notes

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Description** |
| 0.1 | 6/20/18 | Karthik Sarvepalli | Initial draft |
|  |  |  |  |
|  |  |  |  |

## Introduction

### Scope

PhiLance is a platform that individuals and organizations can use to:

* Launch their own social impact project, no matter how big or small that project is (e.g. helping homeless people in your city or finding a wheelchair for a handicapped person in your neighborhood)
* Recruit volunteers and/or freelancers to work on that project. Your project team could comprise of members who live across your street or across the world.
* Manage the project using project management tools provided in the PhiLance platform
* Raise funds for their projects, if required

### Not in Scope

The following items are not considered in scope for this document:

* Physical delivery of the actual project

### General Description

<<Description of the Project>>

Specifications in this document are defined in the following areas:

* Level 1 Business Process Flows - First level process flows that define the user to system and system to system interaction.
* Use Cases - List of the specific user actors relevant to PhiLance platform. List and description of each use case where an actor interacts with the system.
* Functional Requirements - Detailed functional requirements for each use case
* Other Requirements - Requirements for all non-functional areas including legal, operational, data management, customer care, performance, and security
* System Architecture - This broadly covers the areas of system architecture diagrams, interfaces, failure modes between interfacing systems, and list of required APIs

### Guiding Principles

* << Guiding Principals>>

## I/O

* Mobile App
* Web Portal

## Information

Not Applicable

## Functional Requirements

Functional requirements and used cases are categorized using the following criteria:

Release

1. Phase1: Functionality is required for the PhiLance Minimum Viable Product.
2. Phase2: Functionality is required for second launch
3. Phase3: Future Phases: Functionality reserved for future phases beyond Phase 1 and Phase 2

Priority:

1. Required: Functionality is required for PhiLance MVP
2. Desired: Functionality is desired, but not required, for PhiLance MVP
3. Nice to have: Functionality is optional for PhiLance MVP

### Use Cases

#### List of PhiLance Roles/Actors

The following roles serve as the main actors in the defined use cases:

|  |  |  |
| --- | --- | --- |
| **Actor** | **Description** | **Use Cases in Philance** |
|  | | |
| Sponsor | Sponsor is someone who funds a project and is interested in knowing the project outcomes. | * Allocate funds to projects * Oversee the project status * Approve project closures * View funds remaining * View Funds allocate * Approve resources * Export project reports |
| Project Owner | Responsible for bringing in the business case and requests for sponsorship for the project. He may be someone who is benefitted directly or indirectly in completing this project. | * Bring in the need/projects into PhiLance * Find Sponsors * Reach out to sponsors for funding * View project status * View project outcomes |
| Resource | Responsible for executing and completing the tasks assigned to him on a signed-up project. A resource can be anyone from Project Manager or Developer or Tester or anyone executing an assigned task. | * Review tasks assigned * Work on the assigned tasks * Collaborates with team * Provide updates on assigned tasks |
| User | User is just a person/account in the system. He can be a Sponsor and/or Owner and/or Resource in a project. | * Login to system * Create projects as project owner * Search for projects * Apply for projects * Message other users * Take up different roles in a project |
|  |  |  |

### PhiLance

#### Use Cases

##### Use Case: Create Account

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Description** | | |
| **Use Case Number** | UC – 6 | | |
| **Use Case Name** | Create Account | | |
| **Use Case Description** | As a user, I want to sign up in order to create new account on Philance | | |
| **Primary Actor** | User | | |
| **Secondary Actors** |  | | |
| **System Actors** | Philance Web/App | | |
| **Preconditions** | 1. Philance system is up and running | | |
| **Basic Flow** | 1. Volunteer navigate to Philance website 2. Volunteer clicks on Sign Up button 3. Volunteer provides all the required details on the account creation page 4. Volunteer clicks on Submit to create the account. | | |
| **Alternative Flows** | 1. User must be able sign up using Google/Facebook authentication 2. When a user signs up through Google/Facebook authentication, user profile information must be extracted from Google/Facebook and populate on Philance Sign Up page 3. When a user signs up through Google/Facebook authentication, no additional validation email must be sent 4. For Google authenticated account, when user clicks on Sign Up, user must be sent a welcome email | | |
| **Exception Flows** | 1. Provided email id is invalid 2. Google/Facebook authentication fails 3. Provides invalid data 4. Provided user already exists | | |
| **Post Conditions** | 1. A user is created in Philance system. 2. Welcome email is sent to the user | | |
| **Logical Sequence Diagram** |  | | |
| **Functional Requirements** | 1. User must be able to sign up by proving email address 2. When a user signs up by proving email address, email id must be validated for duplicate accounts as soon as the user tabs out the email field 3. User must be asked to confirm the email id on the Sign Up page 4. When a user signs up by proving email address, user must be required to setup a password 5. Password must be at least 8 char and have one uppercase, one lowercase, one special char (@,#,! or $) and one number 6. When a user provides all the required details and clicks on sign up, user must be shown a message stating “Pending Activation, please complete the last step by clicking on the Activation link sent to <User’s email>” 7. When a user tries to sign up using an email address that is pending activation, user must be shown a message stating that the account is pending activation 8. When a user tries to sign up using an email address that is pending activation, user must be given option to resend activation email 9. When a user tries to sign up using an email address that is pending activation, user must be given option to edit email address and profile information 10. If user changes his email address while the account is pending activation, system must send a new activation email to the newly provided email address 11. If user changes his email address while the account is pending activation, system must expire his previous activation link 12. When the user clicks on the Activation link sent to this email, system must complete the activation and display an activation complete message 13. Upon completing the activation process, user must be sent a welcome email | | |
| **Release** | 1 | **Priority** | High |
| **UI Use Case Number** |  | **API Use Case Number** |  |

##### Use Case: Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Description** | | |
| **Use Case Number** | UC – 7 | | |
| **Use Case Name** | Login | | |
| **Use Case Description** | As a user, I want to login in order to use functionality of Philance | | |
| **Primary Actor** | User | | |
| **Secondary Actors** |  | | |
| **System Actors** | Philance Web/App | | |
| **Preconditions** | 1. User account already exists in Philance | | |
| **Basic Flow** | 1. User go to the Philance Web portal/mobile app 2. User clicks on Login 3. User provides email address & password or user clicks on Google/Facebook authentication 4. User clicks on Login | | |
| **Alternative Flows** | 1. N/A | | |
| **Exception Flows** | 1. Invalid password 2. Account does not exist | | |
| **Post Conditions** | 1. User is logged into Philance and can use the functionality | | |
| **Logical Sequence Diagram** |  | | |
| **Functional Requirements** | 1. User must have the ability login to Philace by providing email address and password used for account creation 2. User must be able login to Philance using Google/Facbeook authentication 3. User must be able to use Google Authentication even if it was not used at the time account creation.    1. System must map email IDs and log the user to his account and not create a new account 4. When user types his email address and tabs out the field, system must validate the account and inform the user if the account does not exist 5. If the account does not exist, system must inform the user that the ‘account does not exist, retry with the correct email id or Sign Up’ 6. When the password provided to login incorrect, system must inform the user that the password provided is incorrect and request to retry 7. System must request the user to ‘Reset Password’ after 3 consecutive incorrect attempts 8. On the 4th consecutive incorrect attempt, system must notify the user that the account will be locked after one more attempt 9. On the 5th consecutive incorrect attempt, system must lock the user and request the user to use ‘Reset Password’ option 10. Incorrect login attempts must be counted across all channels(web/app) & sessions 11. User must land the user on the Philance home page after successful authentication | | |
| **Release** | 1 | **Priority** | High |
| **UI Use Case Number** |  | **API Use Case Number** |  |

##### Use Case: Browse Project

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Description** | | |
| **Use Case Number** | UC – 8 | | |
| **Use Case Name** | Browse Project | | |
| **Use Case Description** | As a user, I want to browse existing projects in order to know what projects are listed on Philance | | |
| **Primary Actor** | User | | |
| **Secondary Actors** |  | | |
| **System Actors** | Philance Web/App | | |
| **Preconditions** | 1. User is logged into Philance | | |
| **Basic Flow** | 1. User clicks on Find Projects option 2. User inputs details using basic and advanced search options available to search projects 3. User sees the list of projects matching the search criteria | | |
| **Alternative Flows** | 1. NA | | |
| **Exception Flows** |  | | |
| **Post Conditions** | 1. User navigates to the projects | | |
| **Logical Sequence Diagram** |  | | |
| **Functional Requirements** | 1. User must have an option to find projects posted on Philance 2. Once the user inputs keywords and clicks on Find Projects, user must land on the Find Projects page and see advanced search option, as well as the projects matching the search criteria 3. If user clicks on Find Projects without proving any text, user must land on Find Project page and all projects must be listed 4. User must be given option to perform advanced search 5. Advanced search must include option to search by    1. distance from a given location    2. keywords/Hashtags    3. Skills    4. Impact Category    5. Resource Type    6. Project Status    7. My Project –(Must give ability to search all project created or worked by the user)- Sponsor/Owner/Volunteer 6. All the advanced search options must be optional 7. Search results must be sorted by the date posted 8. Search result must show each projects title, status and brief intro (2 lines) 9. If a user clicks on any project displayed the results window, user must be taken to the project details page 10. User must see an option to go back project results page only when the user goes to project details from find projects page 11. User search criteria and search results must be retained 12. All the projects opened by the user from the search results must be differentiated from projects unopend by color 13. User must be provided an option to choose number of results per page in the search results section | | |
| **Release** | 1 | **Priority** | High |
| **UI Use Case Number** |  | **API Use Case Number** |  |

##### Use Case: Browse Project

Template

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Description** | | |
| **Use Case Number** | UC – <Use Case Number from Github> | | |
| **Use Case Name** |  | | |
| **Use Case Description** |  | | |
| **Primary Actor** |  | | |
| **Secondary Actors** |  | | |
| **System Actors** |  | | |
| **Preconditions** |  | | |
| **Basic Flow** |  | | |
| **Alternative Flows** |  | | |
| **Exception Flows** |  | | |
| **Post Conditions** |  | | |
| **Logical Sequence Diagram** |  | | |
| **Functional Requirements** |  | | |
| **Release** | 1 | **Priority** | High |
| **UI Use Case Number** | Use Case numbers from Github | **API Use Case Number** | Use Case numbers from Github |

## Architecture and Flows

### System Architecture Diagram

<<<<<<<<<<<<<<<<<<<<<<< System Architecture Diagram>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

### Description of Interfacing Systems

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< Details of Third Party Integrations >>>>>>>>>>>>>>>>>>>>>>>

#### Future System Integration

## Other Requirements

### Operational

### Legal Considerations

### Tax Considerations

## Performance Specifications

The system must provides a scalable system to handle various business models and use cases associated with managing services on a global scale. As electronic order capture becomes more prevalent from sources such as web portals and mobile applications, system availability becomes a key metric for the Philance system.

|  |  |  |
| --- | --- | --- |
| **ID** | **Functionality** | **Requirement** |
| Performance - 1 | Availability | System availability must exceed 99.95% |
| Performance - 2 | Availability | API response times from the Philance must be under 1 second |
| Performance - 3 | Processing | The system must support atleast 1000 users and must be scalable |

## Security and Integrity Requirements

This section contains high level security and integrity capabilities that are often provided by SaaS platforms. Many of these elements reflect best practices across the industry, but specific regulations for items such as data privacy and storage are also addressed. These policies will also define the audit and review process the vendor must follow to demonstrate compliance.

### Infrastructure Security

### Product Security

### Corporate & Personnel Security

### Compliance & Privacy

## Appendix

### Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
|  |  |
|  |  |