**Product Requirements**

| **Team** | <Your Team’s Account and Name in Black> |
| --- | --- |

***Note:***

* ***This is a “living document”, meaning its content will change with the implementation of the project. Use it to capture key project requirements and make sure that your product features match the requirements exactly – if you wish to add any features, they must be added first to the requirements. The requirements document, and all changes to it, must be approved by the customer (instructor).***
* ***You are strongly encouraged to continue updating this document with clarifications and details. Clarifications should be indicated [in square brackets].***
* ***Remove this text and the descriptive paragraphs in each section stating what to do before you add this to your repository or turn it in to your instructor.***
* ***You should replace the placeholder text with text that is relevant to your project. The current text is just for an example.***

# **Brief problem statement**

Traditional bookstores face significant challenges in adapting to digital trends and meeting the expectations of modern consumers. The absence of a robust and user-friendly software solution tailored to bookstore operations hampers their ability to efficiently manage inventory, engage customers, and streamline sales processes. Existing point-of-sale systems lack integration with online platforms, hindering the seamless transition between physical and digital sales channels. There is a critical need for a comprehensive software solution that addresses these pain points, providing bookstores with a unified platform to manage inventory, enhance customer interactions, and bridge the gap between offline and online sales channels.

***Replace this text and the instructions below with your statement in black.***  
(2-5 lines describing the problem being addressed. Note that even if you are simply restating what is already in the needs document, you must rephrase it in your words. This gives an opportunity for the customer to identify and provide feedback on differences in interpretation, if any).

# **Stakeholders**

# ***Replace this text and the instructions below with your list in black.***

**Investiny Corp. Board of Directors** – oversee the projects funding and expenses. Have vested interest in the proven success of the product but are not involved in the planning and execution.

**Investiny Corp. Product Owner** – will act as principle representative for Tool Share product needs. He/she champions the product with the Board of Directors, helps facilitate product decisions and has the ultimate say on when and what features should be released.

**Software Engineering Team** – is responsible for the day-to-day operations and coordination of all aspects related to the software product's life-cycle. This include, among others: planning and delegation of team roles and responsibilities; elicitation and clarification of requirements; analysis and design; implementation, testing and release of all software components.

**Beta Testing Team** – represent the target user base for Tools Share. Will be available in later phases of the project to conduct acceptance testing and provide feedback on product release.

# **Users profile**

***Replace this text and the instructions below with your statement in black.***(Identify who will be using the system, in what mode, and their profile in terms of familiarity with using computers and such software. If taking material from needs document, then rephrase in your words).

The target user must:

- have basic experience using computers and browsing the internet. Has filled out online forms or surveys and may have purchased or sold a product.

- have a computer with access to the internet

- have an interest in improving their community by lending or borrowing items of common use with others living near by

- is willing to share information such as home address and contact information

# **System requirements**

***Replace this text and the instructions below with your statement in black.***(1 or more lines identifying the system requirements for your solution. If you require particular languages and libraries, list them as well).

At a high-level this project will be source controlled in SVN, run on Django using python, sqlite and needs to be compatible with the latest browsers.

Although the application needs to be accessible through the internet, deployments and demonstrations for this phase of the project will take place within the RIT Software Engineering environment. To this end, you must understand and document the target platforms from the perspective of the client browser as well as that of the server. Make sure to capture versions or software dependencies, programming languages and hardware specifications that are available for your use and proceed only after you document and confirm these with the customer.

# **Feature requirements (user stories)**

***Read the instructions below and fill in the table. Delete all the blue text before adding this to your repository or turning it in to your instructor.***  
(This is a numbered list of user stories that are the features of the system to be implemented. Each user story is an operation that the user can perform on/with the system. For each user story, provide a fairly detailed description so you know what to build and so you can write a test case to demonstrate that your system provides that feature. For each user story, you will identify (during release planning) the release in which it will be implemented: R1, R2 or R3. Typically, your system will have 10-20 stories or features, but feel free to add more table rows if you decide to use finer-grain stories).

The following list of user stories is neither final nor comprehensive. You must consider it your responsibility to maintain it's relevance, clarify any misunderstandings and keep it up-to-date. Any changes must be discussed with the Product Owner for approval.

| **No.** | **User Story Name** | **Description** | **Release** |
| --- | --- | --- | --- |
| **1** | Registration | Registrant shall provide personal information and preferences to the System upon registering and becoming a User. (for example: Last name or Pickup Arrangements)  System will create a new Share Zone upon registration of User with previously unregistered zipcode  Upon registration System will associate user with an already created Share Zone based on zipcode | R1 |
| **2** | Account Management - change personal info | User shall be able to change personal information after registration (for example: change address).  System will warn User if changing of community and effect any necessary changes to the item list. | R1 |
| **3** | Account Management - change user preferences | User shall be able to change preferences after registration (for example: email reminder frequency, pickup arrangements) | R1 |
| **4** | Community Shed creation | User shall be able to create a Shed for his Share Zone.  A Shed has a physical location where tools from several Users can be stored.  Shed creator becomes Coordinator of said shed. | R1 |
| **5** | Tool Management – registration | User can register a tool by providing items information. (for example: Picture, Category, Name, Description, Special instructions, etc.)  System will require a unique field to distinguish between similar tools (for example: two laser blasters but mine has “Buzz A 113” edged on its handle)  User must accept default Pickup Arrangement (set during User registration) or select a different arrangement for this tool (for example: “Please knock on my door” vs. “Let's make additional arrangements via email”) | R1 |
| **6** | Sharing – from Home | User can set a previously registered tool to be shared from home.  Tool cannot be shared from two places simultaneously  *User must physically have possession of tool prior to sharing it.*  User can change a previously registered tool | R2 |
| **7** | Sharing – from Community Shed | User can set a previously registered tool to be shared from a Community Shed.  Tool cannot be shared from two places simultaneously.  *User must physically move the tool to Shed prior to being allowed to share it.*  Shed Coordinator can, from time to time, verify and change status of tools in Community Shed (for example: a tool was returned but never marked as present or tool has gone missing) | R1 |
| **8** | Sharing – change location of tool | User can change a previously registered tool to be shared elsewhere.  System will prevent relocation when tool has an unresolved future reservation. | R2 |
| **9** | Sharing – change availability | User can availability of a tool. (for example: on black out dates when he/she will be using the tool, or be away/unwilling) | R1 |
| **10** | Sharing – approving a borrower | System will notify User upon a request to borrow his/her tool.  User owner of tool must approve the borrower when his/her tool is not at a Community Shed.  System will notify borrower of decision by owner and update tools status if necessary. | R1 |
| **11** | Tool Listing – availability | User can request from System a list of all tools and their availability. List shall be sorted by System.  Only tools in User's Share Zone will be visible | R1 |
| **12** | Borrowing - request | User can select to borrow a tool for specified days provided tool is available. Borrower can add a message to request.  User owner of tool will need to “Approve” or “Reject” the request to borrow unless tools is in a Community Shed were it is always automatically approved.  Upon approval, System will create Reservation with requested dates and update availability of a borrowed tool.  Upon rejection, owner of tool must provide a small reason for “Reject”.  System will send notification email to borrower or lender including reason or message where applicable. | R1 |
| **13** | Borrowing – arrangements | Location for user sharing a tool. | R1 |
| **14** | Borrowing - returning | Borrower will return tool pickup location unless otherwise agreed with tool owner.  *Borrower will notify the System that tool has been “Returned”*  User owner of tool will need to notify/acknowledge to the System that tool has been “Returned” unless request to borrow was made from a Community Shed were it will be acknowledged by Shed Coordinator.  System will update availability of a returned tool. | R1 |
| **15** | Tool Management – deactivation and reactivation | User can deactivate a tool they previously added to the System. Neither the tool nor its reservations are shown by default, but the system still retains the information about the tool. A tool may be re-activated at any time.  System will require confirmation in case of possible conflicts.  System will notify borrowers of tool being deregistered and update relevant information to disable availability of tool. | R2 |
| **16** | Tool Management – Status | User owner of tool can request from System a list of registered owned tools and their status. | R2 |
| **17** | Community Statistics | User can request a list of statistics for their specific Share Zone. Upon request System will verify association and display such listing.  Statistics that may be included:  - most active lenders  - most active borrowers  - most used tools  - most recently-used tools | R2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Use case diagram**

***Read the instructions below and fill in the content. Delete all the blue text before adding this to your repository or turning it in to your instructor.***

***Draw the UML use case context diagram for the system. As you update, make sure the use cases shown in the diagram correspond to the user stories described in the previous section and vice versa. A partial sample breakdown is provided.***

***Registrant***

***Registration***

***User:Owner***

***Account Management***

***Tool Management***

***Community Shed creation***

***Tool Listing***

***Tool Sharing***

***Community Statistics***

***User:Borrower***

***Account Management***

***Community Shed creation***

***Tool Listing***

***Tool Borrowing***

***Community Statistics***

***User:Shed Coordinator***

***Tool Listing***

***User:Admin***

***User Administration***

***System Administration***

**Use case description**

***Delete all the blue text and fill-in the template before adding this to your repository or turning it in to your instructor.******NOTE: In order to avoid confusion, it is STRONGLY encouraged to complete all use cases prior to implementation. Use the template and example provided as a guide.***

***Sample Template:***

| **Use Case Number:** | *UC-XX (Replace XX with a number)* |
| --- | --- |
| **Use Case Name:** | *Enter the name of Use Case* |
| **Overview:** | *Describe the purpose of the Use Case and give a 1-2 line description. This could be the same as the description provided for the user story.* |
| **Actor(s):** | *List all actors that participate in this Use Case.* |
| **Pre condition(s):** | *Enter the condition that must be true when the main flow is completed.* |
| **Scenario Flow:** | *Main (success) Flow: Steps should be numbered.* |
|  | *Alternate Flows: Include the post condition for each alternate flow if different from the main flow.* |
| **Post Condition:** | *Enter the condition that must be true when the main flow is completed.* |

***Sample Use Case:***

| **Use Case Number:** | *UC-01* |
| --- | --- |
| **Use Case Name:** | *Registration* |
| **Overview:** | *Registrant shall provide personal information and preferences to the System upon registering and becoming a User.* |
| **Actor(s):** | *Registrant* |
| **Pre condition(s):** | *- System has been setup and configured.*  *- System is running and open for registrations.*  *- Registrant has accessed website via URL* |
| **Scenario Flow:** | *Main (success) Flow:*   1. *Registrant selects option to register* 2. *System requests personal information* 3. *Registrant provided personal information.* 4. *System verifies required information is provided.*  * *If information is invalid System displays message. Return to Step 2*  1. *System requests user's preferences* 2. *Registrant provides user preferences* 3. *System verifies required information is provided.*  * *If information is invalid System displays message. Return to Step 5*  1. *If user's zipcode is new to the System, new ShareZone is created* 2. *System will associate user with corresponding ShareZone* 3. *System displays confirmation of registration* |
|  | ***Alternate Flows:***  *After Step(2) System will display the option to Cancel the registratio process. The following steps would occur:*   1. *Registrant selects option to cancel during registration* 2. *System requests confirmation to cancel* 3. *Registrant confirms intent* 4. *System returns to main screen* |
| **Post Condition:** | *Registrant did not complete registration. System does not store Registrant's information.* |