

Product Requirements

Team : g262-01a (The Squad)

Brief problem statement

We represent a funding group (Investiny Corp.) chartered to create applications for the benefit of communities all around the country. The product we envision is called Tool Share. At its core, Tool Share is meant to enable neighbors in a community to be able to share items of common use. The successful implementation should make it easy for anyone wanting to participate to register and be able to share or borrow items.

The Tool Share product is intended to improve communities by providing an easy mechanism for sharing items between neighbors. We envision this to primarily consist of tool exchanges but the implementation must be easily extended to include items of different kinds.

We want a product whose emphasis is on ease of use, whose navigation is straightforward and where the status of items and users is clearly displayed.

We are revolutionizing the tool shed and launching it into the next generation. Our vision is to provide an experience for tracking tools that you share with friends and family in an fluid and intuitive environment.

Stakeholders

Investiny Corp. Board of Directors – Oversee the project and are in charge of funding and making sure the project is being kept on the proper path.

Investiny Corp. Product Owner – The final decision maker for the project. He/she will determine the specifics of the project and how they should function for the companies overall final project view.

Software Engineering Team – Will do the actual production of the product. They will consult the Investiny board of directors when discussing specific issues and how to advance with the project. All of the grunt work from building to managing the project even after it is complete.

Beta Testing Team – The team in charge of testing the product in its different phases of completion. They will have all of the product available to them and then they will be able to try it and document what works and what doesn't work. After the documents are filled out they will be given to the software engineers to resolve the problems that were discovered.

Users profile

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

The target user is:

- in a nice suburban community where tools can be shared
- a person who has interest in tool sharing or borrowing
- a person who might want to be the admin of a shed

System requirements

At a high-level this project will be source controlled in SVN, ran on Django 1.6.6 using python3, sqlite and needs to be compatible with the latest browsers. A testing framework will be used to run nightly regression tests and will be initiated by a cron script running on an external server. For back end debugging a SQL GUI will be used such as SQLiteBrowser in order to view and edit the user database. For front end the Bootstrap framework will be used complemented by Javascript where needed.

Feature requirements (user 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.

NOTE: We expect back-end functionality for each user story to be complete by R1. However, our front-end for R1 will be very minimal and limited to user registration essentially.

No	User Story Name	Description	Release
1	Registration	Registrant puts in name and email. System emails him a temporary password and a confirmation link. new user enters new password and fills in personal information and submits.	R1
2	Account Management - change personal info	User types in their password prior to being allowed to change personal information. After verifying user can change whatever they like and hit submit. Changing phone number requires verification code to be confirmed.	R2
3	Account Management - change user preferences	Preferences specifically for their account and specific tools can be changed at any time.	R2
4	Community Shed creation	A physical shed location where users can put all their tools in one user that created it becomes the admin of the location but he can set others as admins as well.	R2
5	Tool Management – registration	User creates a new tool and fills in description information about it as well as unique identifiers to help tell the difference between similar things. System can generate a QR code if user wants to stick it onto the tool prior to taking a picture of the tool for the system.	R2
6	Sharing – from Home	Owner of the tool can change the share location of the tool to be his own home rather than a community shed. This can not be changed when the tool is currently lent	R2
7	Sharing – from Community Shed	The tool share location can be changed at any time but when the tool is lent to someone who is not the owner. Tool cannot be shared from two places simultaneously.	R2

		<p><i>User must physically move the tool to Shed prior to being allowed to share it.</i></p> <p>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)</p>	
8	Sharing – change location of tool	<p>Owner can change the location of a tool that they have in their possession</p> <p>System will prevent relocation when tool has an unresolved future reservation.</p>	R2
9	Sharing – change availability	User can change the availability of a tool of theirs or change dates that the tool will not be available for other people to use.	R2
10	Sharing – approving a borrower	<p>A message will pop up notifying the user that another user wants to borrow their tool. The user can choose to agree to it or deny.</p> <p>The system will then send the approval or rejection back to the requester.</p>	R2
11	Tool Listing – availability	User requests for a list from the system. The system will then find all tools and then display them and their availability.	R2
12	Borrowing - request	<p>User selects what tool they want to borrow and the system offers time options such as indefinite or whatever the max borrow time is set for that specific persons tool.</p> <p>The system will notify both borrower and lender about the request and will notify the borrower when the lender makes a decision.</p>	R2
13	Borrowing – arrangements	<p>Depending on specific preferences for tool, Lenders address is given to borrower and a specific time.</p> <p>Lender takes a picture of the tool and uploads it as last known condition</p>	R2
14	Borrowing - returning	<p>Borrower will return the tool to the owner or the shed (whichever it was borrowed from) and take a picture of it.</p> <p>They will then be required to post the picture they took of it's current state.</p> <p>The owner of the tool will then have to verify that the tool has been returned and in proper condition.</p> <p>System will update availability of a returned tool.</p>	R2

15	Tool Management – deregistration	<p>Owner of a tool can deregister a registered tool that they have out so that it will not be shown for other users or on the availability list</p> <p>System will require confirmation in case of possible conflicts.</p> <p>System will notify other users that the tool has been removed as requested by the owner.</p>	R2
16	Tool Management – Status	<p>User can request from the system a list of their personal tools and their availability.</p> <p>It will show tools that they currently are borrowing and the tools that they already own.</p>	R2
17	Community Statistics	User can request from the system some statistics of all tools, who borrows tools the most, who lends tools the most, recently used tools, and most popular tools	R2

Use case diagram

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 Cases:

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):	<ul style="list-style-type: none">- System has been setup and configured.- System is running and open for registrations.- Registrant has accessed website via URL
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none">1. Registrant selects option to register2. System requests personal information3. Registrant provided personal information.4. System verifies required information is provided.<ul style="list-style-type: none">• If information is invalid System displays message. Return to Step 25. System requests user's preferences6. Registrant provides user preferences7. System verifies required information is provided.<ul style="list-style-type: none">• If information is invalid System displays message. Return to Step 58. If user's zipcode is new to the System, new ShareZone is created9. System will associate user with corresponding ShareZone10. System displays confirmation of registration
Alternate Flow	<p>After Step(2) System will display the option to Cancel the registration process. The following steps would occur:</p> <ol style="list-style-type: none">1. Registrant selects option to cancel during registration

	<ol style="list-style-type: none"> 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.

Use Case Number:	UC-02
Use Case Name:	Account Management
Overview:	User updates his or her personal account information.
Actor(s):	User - Owner
Pre condition(s):	<ul style="list-style-type: none"> - User has registered for an account previously - User is logged in to ToolShare - User accesses the account management link
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1. User selects option to update his or her account. 2. System verifies that user is logged in. 3. System redirects user to Account Management page. 4. User updates the information that he or she needs to. 5. System sanitizes entered text and checks for proper format. <ul style="list-style-type: none"> • If information is invalid System displays message. Return to Step 4. 6. If user's zipcode is new to the System, new ShareZone is created 7. System will associate user with corresponding ShareZone 8. System displays confirmation of updated account information.
Alternate Flow:	In step 5 we have an alternate flow for updating personal account information. If any of the info entered is invalid the System will display a message to the user and send he/she back to step 4.
Post Condition:	User's account is updated successfully and changes are saved in database.

Use Case Number:	UC-03
-------------------------	-------

Use Case Name:	Tool Management
Overview:	User views his or her tools and all relevant information about the tools. Has the option to delete and edit tools.
Actor(s):	User - Owner
Pre condition(s):	- User is logged into ToolShare
Scenario Flow	<p>Main (success flow):</p> <ol style="list-style-type: none"> 1. User accesses the tool management link 2. System displays all of the user's tools in a list 3. User has the option to select tools and delete/edit <ul style="list-style-type: none"> - If "delete" selected, tool is deleted from the tool shed permanently - If "edit" selected, system brings you to a page to edit tool info 4. System displays confirmation of updated tool(s).
Alternate Flow:	On step 3, if "delete" is selected the System will refresh the list of tools with that tool deleted. If "edit" is selected the System will bring the user to a simple page with a form where he can edit the tool info. Once "save" is clicked, the user will be redirected back to the page with the list of tools.
Post Condition:	User's tools are updated and changes are displayed across the entire website.

Use Case Number:	UC-04
Use Case Name:	Community Shed Creation
Overview:	User decides he wants to create a new community shed for tool sharing within his or her ShareZone.
Actor(s):	User - Owner
Pre condition(s):	- User is logged into ToolShare
Scenario Flow	<p>Main (success flow):</p> <ol style="list-style-type: none"> 1. User clicks "Create new community shed" 2. System brings user to a page to fill out shed name. 3. User fills in information 4. New shed is created and stored in the database

	5. System verifies creation of the shed with a success message.
Alternate Flow:	If in step 3 the user fills in an address that can not be validated, the System will display an error message and keep the user on the shed creation page.
Post Condition:	A new community shed is created and the system displays a success message.

Use Case Number:	UC-5
Use Case Name:	Tool Listing
Overview:	Tool Listing will show the user all the tools that are available for them to borrow
Actor(s):	User -> System
Pre condition(s):	User is logged into the URL
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1. User requests info from the system 2. System sorts the list of tools 3. Display the list of tools and their availability 4. User's own tools will only be displayed if they are in the user's share zone
Alternate Flow:	No alternate flow - tools are simply displayed to the user. Therefore there are no errors or special cases to deal with. If the user has no tools the page will simply display no tools.
Post Condition:	System is displaying a list of tools to the user and their availability

Use Case Number:	UC-6
Use Case Name:	Tool Sharing
Overview:	Tool sharing will allow the user to share his or her tool with other users in the system
Actor(s):	User -> System
Pre condition(s):	<p>User is logged into the URL</p> <p>To share a tool they must previously have one in their tool managment</p>

Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1. User request to share tools from the system 2. If user has a tool from his home that was previously registered they can change the status of the tool 3. they can try to borrow other user's tools by having the system send a request to the other user 4. the user can change the location of a borrowed tool if it has changed locations 5. the user can add the tool to the community shed rather than their home tools (this requires the owner to deliver the tool to the community shed) 6. if the user receives a borrow request from another user they can accept or decline it
Alternate Flow:	<ol style="list-style-type: none"> 1. If the user tries to borrow a tool that is already being borrowed the system will send an error to the user 2. If the user tries to change a tool's status without having it in possession
Post Condition:	System will update changed tool status's, their location if the tools sharing information is changed

Use Case Number:	UC-7
Use Case Name:	Community Statistics
Overview:	User can request statistics from the systems
Actor(s):	User -> System
Pre condition(s):	User is logged into the URL
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1. User requests info from the system

	2. System gets the statics of lenders and tools 3. Display the list of statics to the user
Alternate Flow:	1. If the user hasn't shared any tools yet the system will display a message that says you haven't shared anything yet
Post Condition:	System is displaying a list of tools to the user and their availability

Use Case Number:	UC-08
Use Case Name:	Account Management
Overview:	User updates his or her personal account information.
Actor(s):	User - Borrower
Pre condition(s):	<ul style="list-style-type: none"> - User has registered for an account previously - User is logged in to ToolShare - User accesses the account management link
Scenario Flow:	Main (success) Flow: 1. User selects option to update his or her account. 2. System verifies that user is logged in. 3. System redirects user to Account Management page. 4. User updates the information that he or she needs to. 5. System sanitizes entered text and checks for proper format. 6. If user's zipcode is new to the System, new ShareZone is created 7. System will associate user with corresponding ShareZone 8. System displays confirmation of updated account information.
Alternate Flow:	On step 5, if information is invalid System displays message. Return to Step 4.
Post Condition:	User's account is updated successfully and changes are saved in database.

Use Case Number:	UC-09
Use Case Name:	<i>Community Shed creation</i>
Overview:	User creates new community shed for their specific area.
Actor(s):	User - Borrower
Pre condition(s):	<ul style="list-style-type: none"> - User has registered for an account previously - User is logged in to ToolShare - User accesses the new community shed link
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1. User inputs the name and location of their new shed group 2. If name not taken for specific area community shed group is created with user as first member. 3. Page requesting user to invite people to his community shed group appears. 4. Once user enters invitee emails and hits submit system checks the emails across current accounts and sends invites through system for existing users and new emails requesting to join for new users. 5. Page notifies user of invitations sent.
Alternate Flow:	On step 4 if an email does not exist on ToolShare or if an email is in an invalid format the System will display an error message and keep the user on the same page to fix his mistake.
Post Condition:	New tool share group added to system database with original user as only member until others accept the invitations.

Use Case Number:	UC-10
-------------------------	-------

Use Case Name:	Tool Listing
Overview:	The user can view all tools available for them to borrow
Actor(s):	User - Borrower
Pre condition(s):	User has registered and logged in
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1.) User requests information from system 2.) System sorts list of tools in area of user 3.) System displays all tools in area of user and their availability
Alternate Flow:	No alternate flow. Tools are simply listed.
Post Condition:	System is displaying a list of tools to user and their availability

Use Case Number:	UC-11
Use Case Name:	Tool Borrowing
Overview:	The user can borrow an available tool of their choosing from another user
Actor(s):	User – Borrower, User - Owner
Pre condition(s):	User has registered and logged in
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1.) User requests to borrow a tool from system 2.) System sends request to tool owner to approve 3.) Tool owner approves request 4.) System grants permission to borrower
Alternate Flow:	1.) User requests to borrow a tool that is unavailable, system sends error to user

	2.) User's request to borrow tool from owner is denied
Post Condition:	User is able to successfully borrow tool of their choosing, system updates tool's status to unavailable

Use Case Number:	UC-14
Use Case Name:	User Administration
Overview:	User is being unruly and needs to be punished accordingly.
Actor(s):	User, Admin
Pre condition(s):	<ul style="list-style-type: none"> - User is a current account holder - User has broken ToolShare rules - Staff decide user should be banned.
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1. Admin logs into administration panel 2. Admin goes to user search 3. Admin goes to user's Admin control panel 4. Admin checks tick box to set user as banned.
Alternate Flow:	No alternate flow. If the search yields no results the page will simply display no users.
Post Condition:	User receives a message stating that they are banned upon attempted login.

Use Case Number:	UC-15
Use Case Name:	System Administration
Overview:	New tool category needs to be added
Actor(s):	Admin
Pre condition(s):	<ul style="list-style-type: none"> - Staff determine that there are tools wanting to be shared that do not fit in any category. - There are too many of that tool to leave in "Other"
Scenario Flow:	<p>Main (success) Flow:</p> <ol style="list-style-type: none"> 1. Admin logs into administration panel

	2. Admin goes to categories tab. 3. Admin clicks + 4. Admin enters name and selects tool to make subcategory (if necessary)
Alternate Flow:	No alternate flow.
Post Condition:	New tool category is available for users to list under.

Use Case Number:	UC-16
Use Case Name:	Community Statistics
Overview:	A borrower is checking community statistics.
Actor(s):	User - Borrower, System
Pre condition(s):	User is logged in.
Scenario Flow:	<i>Main (success) Flow:</i> 1. User requests info from the system 2. System gets the statics of lenders and tools 3. Display the list of statics to the user
Alternate Flow:	No alternate flow. Stats are simply displayed on the page.
Post Condition:	User is viewing a page with tools and their availability

Use Case Number:	UC-17
Use Case Name:	Add Tool
Overview:	Shed Coordinator is adding a new tool to the shed listings.

Actor(s):	User - Shed Coordinator, System
Pre condition(s):	<ul style="list-style-type: none"> - A shed must exist - Shed Coordinator user is logged in
Scenario Flow:	<p><i>Main (success) Flow:</i></p> <ol style="list-style-type: none"> 1. User is on shed listing page 2. User clicks add tool button 3. User fills out form and submits it 4. System adds tool to listings db 5. User is redirected to listings page
Alternate Flow:	No alternate flow.
Post Condition:	A new tool is now in the shed listing.

