

BETA TEST PLAN

1. Core Functionalities for Beta Version

Below are the essential features that must be available for beta testing, along with any changes made since the initial Tech3 Action Plan.

Feature Name	Description	Priority (High/Medium/Low)	Changes Since Tech3
Role-based access control	Allows access to core features based on user roles (Owner / Tenant).	High	-
Property management	Property creation, modification and archiving.	High	-
Property dashboard	Invite a tenant and manage lease, manage inventory (rooms and furnitures), see documents and damages.	High	-
Inventory report	Guided and assisted inventory report based on property inventory.	High	-
Image analysis	Analyze images taken during inventory report to auto-generate summary.	High	On server rather than on device
Damage report	Tenant can report damages in a property and a follow-up of the fix is done by the owner.	Medium	-
Documents	Upload and manage documents related to the property (e.g., leases, reports, bills...).	Medium	-
General dashboard	Overview of all properties, leases, messages and damages.	Medium	-
Inventory management	Add rooms and items in the inventory. Pre-made templates to easily create properties.	Medium	-
Import/export inventories	Import and export property inventory data in various formats (e.g., CSV, JSON).	Medium	-
Messaging system	Chat between tenants and owners.	Low	-
Settings	User settings (profile, notifications, accessibility, etc).	Low	-

2. Beta Testing Scenarios

2.1 User Roles

The following roles will be involved in beta testing.

Role Name	Description
Property Owner	User who owns and manages properties (create, edit, delete, invite tenants).
Tenant	User invited to rent a property, with limited access to property details and contract-related features.

2.2 Test Scenarios

For each core functionality, provide detailed test scenarios.

Web scenarios

Scenario 1: Property Creation

- **Role Involved:** Property Owner
- **Objective:** Test the property creation functionality
- **Preconditions:** User is logged in with appropriate permissions
- **Test Steps:**
 1. Navigate to property page
 2. Click on add property button
 3. Fill in property details (address, description, price, ...)
 4. Upload property images
 5. Submit property listing
- **Expected Outcome:**
 - Property is successfully created
 - Property appears in user's property list
 - All property details are correctly saved
 - Success message is displayed

Scenario 2: Property Modification

- **Role Involved:** Property Owner
- **Objective:** Test the modification of an existing property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property page
 2. Choose a property to updated
 3. Click on the drop-down menu at the top right of the property and update the property
 4. Update details (e.g., address, monthly rent, name, etc)
 5. Click "Save Changes"
- **Expected Outcome:**
 - Modifications are saved successfully
 - Updated details (address, rent) and image (if uploaded) appear immediately in the property list
 - Success message "Property updated successfully" is displayed

Scenario 3: Property Archiving

- **Role Involved:** Property Owner
- **Objective:** Test the archiving of a property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property page
 2. Choose a property to archived
 3. Click on the drop-down menu at the top right of the property and delete the property
 4. Confirm deletion in the alert dialog
 5. Return to the properties page and click on the toggle switch to display archived properties
- **Expected Outcome:**
 - Property is archived successfully
 - Property disappears from the active property list
 - Property appears in the archived properties list
 - No errors are displayed
 - Success message "Property archived successfully" is displayed

Scenario 4: Add Document

- **Role Involved:** Property Owner
- **Objective:** Test the add document in a property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
 - A lease exists for the property
- **Test Steps:**
 1. Select a lease (if not current lease)
 2. Click on the "Add Document" button

3. Enter a name for the document
4. Select a PDF file to upload
5. Click on the "Add" button

- **Expected Outcome:**
 - Document appears with the given name
 - Document upload date is displayed
 - PDF can be opened in the browser
 - Success message "Document added successfully" is displayed

Scenario 5: Delete Document

- **Role Involved:** Property Owner
- **Objective:** Test to delete document in a property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
 - A lease exists for the property
 - A document exists in the property
- **Test Steps:**
 1. Select a lease (if not current lease)
 2. Click on the delete icon next to the document
 3. Confirm deletion in the alert dialog
- **Expected Outcome:**
 - Document is no longer visible on the page
 - Success message "Document deleted successfully" is displayed

Scenario 6: Inventory Report Creation

- **Role Involved:** Property Owner
- **Objective:** Test the inventory report creation
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property to which a inventory report will be created
 3. Navigate to the inventory tab
 4. Click on "Add Room" button and fill in the room name
 5. Select a room template from the dropdown menu (e.g. bedroom, bathroom, kitchen, etc.) which will pre-fill common items for that room type
 6. After adding the room, click on "Add Item" button in the room container to add items
 7. Fill in the name and number of items
- **Expected Outcome:**
 - Room is successfully created
 - Success message "Room created successfully" is displayed
 - Item is successfully created
 - Success message "Item added successfully" is displayed

Scenario 7: Update Room

- **Role Involved:** Property Owner
- **Objective:** Test the modification of a room from an inventory
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
 - A room exist in the property
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with a room
 3. Navigate to the inventory tab
 4. Click on the update icon next to the room name
 5. Modify the room name and/or room type in the form
 6. Click on the confirm button to save changes
- **Expected Outcome:**
 - Room name and/or type is updated in the inventory
 - Success message appears confirming the modification

Scenario 8: Update Item

- **Role Involved:** Property Owner
- **Objective:** Test the modification of an item from a room
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
 - A room exist in the property
 - An item exist in a room
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with a room
 3. Navigate to the inventory tab
 4. Click on the item you want to modify
 5. Update the item name and/or quantity in the form
 6. Click on the confirm button to save changes
- **Expected Outcome:**
 - Item name and/or quantity is updated in the room
 - Success message appears confirming the modification

Scenario 9: Delete Room

- **Role Involved:** Property Owner
- **Objective:** Test the deletion of a room from an inventory
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
 - A room exist in the property
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with a room
 3. Navigate to the inventory tab
 4. Click on the delete icon next to the room
 5. Confirm deletion in the alert dialog
- **Expected Outcome:**
 - Room is no longer visible in the inventory
 - Success message appears confirming the deletion

Scenario 10: Delete Item

- **Role Involved:** Property Owner
- **Objective:** Test the deletion of an item from a room
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
 - A room exist in the property
 - An item exist in a room
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with a room and an item in it
 3. Navigate to the inventory tab
 4. Click on the delete icon next to the item
 5. Confirm deletion in the alert dialog
- **Expected Outcome:**
 - Item is no longer visible in the room
 - Success message appears confirming the deletion

Scenario 11: The owner can check damages of his property

Role Involved: Property Owner

- **Objective:** Check if the owner can see the damages of his property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
 - Damage exists in the property

- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with damages
 3. Click on the damages tab
- **Expected Outcome:**
 - The damage are loaded and well displayed
 - There is no bugs

Scenario 12: The owner can set the expected fix date for a damage

Role Involved: Property Owner

- **Objective:** Check if the owner can see the damages of his property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
 - Damage exists in the property
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with damages
 3. Click on the damages tab
 4. Select a damage and click on the intervention date field
 5. Choose a future date from the date picker
- **Expected Outcome:**
 - The damage is updated with the intervention date
 - Success message "Intervention date updated successfully" is displayed
 - There are no bugs

Scenario 13: The owner can mark a damage as fixed

Role Involved: Property Owner

- **Objective:** Check if the owner can mark a damage as fixed
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
 - Damage exists in the property
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with damages
 3. Click on the damages tab
 4. Select a damage and click on the "Mark as Fixed" button
 5. Click "Confirm" in the confirmation dialog
- **Expected Outcome:**
 - Damage status changes to "pending tenant confirmation"
 - Success message "Damage marked as fixed, waiting for tenant confirmation" is displayed

Scenario 14: Invite Tenant

- **Role Involved:** Property Owner
- **Objective:** Test the invite tenant functionality
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
 - Property is available
 - Property is not archived
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property to which a tenant will be invited
 3. Click on the drop-down menu at the top right of the property and add a tenant
 4. Fill in contract details (tenant email, start date of the contract, end date of the contract is optional)
- **Expected Outcome:**
 - Tenant is successfully invited
 - The property badge changes to **invitation sent**
 - The tenant receives an e-mail affiliated with the property to create an Keyz account
 - Success message "Tenant invitation sent successfully" is displayed

Scenario 15: Cancel the Tenant invitation

- **Role Involved:** Property Owner
- **Objective:** Test the cancel invite tenant functionality
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
 - Property has 'invitation sent' status
 - Property is not archived
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with the badge **invitation sent**
 3. Click on the drop-down menu at the top right of the property and click on the "Cancel invitation" button
 4. Click on confirm on the pop up
- **Expected Outcome:**
 - Invitation is successfully canceled
 - The property badge changes to **available**, and all mentions of the tenant disappear from it
 - The invitation email link is deactivated
 - Success message "Tenant invitation canceled successfully" is displayed

Scenario 16: End a Lease

- **Role Involved:** Property Owner
- **Objective:** Test the ability to end an active lease for a property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
 - A current lease exists for the property (occupied)
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with the badge **occupied**
 3. Click on the drop-down menu at the top right of the property and click on the "End lease" button
 4. Click on confirm on the pop up
- **Expected Outcome:**
 - Property status changes to **available**
 - Tenant information is no longer displayed in the property
 - Success message "Lease ended successfully" is displayed

Scenario 17: See owner profile

- **Role Involved:** Property Owner
- **Objective:** Test if the owner can see his profile informations
- **Preconditions:**
 - User is logged in with appropriate permissions
- **Test Steps:**
 1. Navigate to settings page
 2. Check the profile informations
- **Expected Outcome:**
 - The informations are well displayed and good

Scenario 18: Change the profile picture

- **Role Involved:** Property Owner
- **Objective:** Test if the owner can change his profile picture
- **Preconditions:**
 - User is logged in with appropriate permissions
- **Test Steps:**
 1. Navigate to settings page
 2. Click on the user icon
 3. Select a new profile picture from the device
- **Expected Outcome:**
 - Profile picture is updated

- Success message "Picture updated successfully" is displayed

Scenario 19: Change the app language

- **Role Involved:** Property Owner
- **Objective:** Test if the owner can change the language of the app
- **Preconditions:**
 - User is logged in with appropriate permissions
- **Test Steps:**
 1. Navigate to settings page
 2. Click on the desired language in the language section
- **Expected Outcome:**
 - The app change his language
 - The UI is the same, no bugs

Scenario 20: View Messages and Contacts

- **Role Involved:** Property Owner
- **Objective:** Test if the owner can view their messages and contacts
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system and is occupied by a tenant
- **Test Steps:**
 1. Navigate to messages page
 2. Search for a contact in the search bar or click on an existing contact to view messages
- **Expected Outcome:**
 - Messages and contacts are loaded correctly

Scenario 21: Change the app language

- **Role Involved:** Property Owner
- **Objective:** Test if the owner can send messages
- **Preconditions:**
 - User is logged in with appropriate permissions
- **Test Steps:**
 1. Navigate to messages page
 2. Select a contact from the list or search for a specific contact
 3. Click on the contact to open the chat
 4. Type a message in the input field or click the attachment icon to send a file
 5. Click the send button
- **Expected Outcome:**
 - Message is sent and displayed in the conversation

Mobile scenarios

Scenario 1: Property Modification

- **Role Involved:** Property Owner
- **Objective:** Test the modification of an existing property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property view
 2. Click on a property and click "Edit"
 3. Update details (e.g., address, monthly rent, name, etc)
 4. Click "Save Changes"
- **Expected Outcome:**
 - Modifications are saved successfully
 - Updated details (address, rent) and image (if uploaded) appear immediately in the property

Scenario 2: Property Archiving

- **Role Involved:** Property Owner
- **Objective:** Test the archiving of a property

- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property page
 2. Click on the property to open it
 3. Click on the moreVert button to open the panel of actions
 4. Click on the delete button
 5. Confirm deletion in the alert dialog
- **Expected Outcome:**
 - Property is archived successfully
 - Property disappears from the active property list
 - Property view is closed
 - No errors are displayed

Scenario 3: Consulting an Inventory Report

- **Role Involved:** Property Owner & Property Tenant
- **Objective:** Test the ability to view an existing inventory report for a property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system with at least one inventory report made (entry or exit)
- **Test Steps:**
 1. Navigate to the property page
 2. Select a property with an existing inventory report
 3. Click on the property on the properties list
 4. Click on the document tab
 5. Click on the document we want to open
- **Expected Outcome:**
 - The inventory report is displayed correctly in the pdf viewer
 - The interface is responsive and loads without errors

Scenario 4: Create an entry or exit Inventory Report

- **Role Involved:** Property Owner & Property Tenant
- **Objective:** Test the creation of a new inventory report
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property page
 2. Select a property
 3. Click on the property on the properties list with a tenant in it or in your own property if you're a tenant
 4. Click on the inventory button
 5. Click on each of the rooms to fill their furnitures forms and the room form
 6. Once each of the room and their furnitures are filled, click on the submit inventory button
 7. Click confirm
- **Expected Outcome:**
 - We can see the newly created inventory report in the documents
 - The inventory report can be added without any issue or bug
 - Each page show no error and can be completed
 - The interface is responsive and loads without errors

Scenario 5: The user can use the AI to evaluate or compare the state of his room or furniture

- **Role Involved:** Property Owner & Property Tenant
- **Objective:** Test the evaluation by AI of the state of the room or the furniture
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property page
 2. Select a property
 3. Click on the property on the properties list with a tenant in it or in your own property if you're a tenant
 4. Click on the inventory button
 5. Click on each of the rooms to fill their furnitures forms and the room form
 6. Once you are on a form of a room or a form of a furniture, upload a picture on the picture carousel
 7. Click on the evaluate or compare button
- **Expected Outcome:**
 - The form is filled with the data of the AI
 - The form values correspond to the image
 - The interface is responsive and loads without errors

Scenario 6: Property Creation

- **Role Involved:** Property Owner
- **Objective:** Test the property creation functionality
- **Preconditions:** User is logged in with appropriate permissions
- **Test Steps:**
 1. Navigate to property page

2. Click on add property button
3. Fill in property details (address, description, price, ...)
4. Upload property images
5. Submit property listing

- **Expected Outcome:**
 - Property is successfully created
 - Property appears in user's property list
 - All property details are correctly saved

Scenario 7: Invite Tenant

- **Role Involved:** Property Owner
- **Objective:** Test the invite tenant functionality
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property to which a tenant will be invited
 3. Click on the drop-down menu at the top right of the property and add a tenant
 4. Fill in contract details (tenant email, start date of the contract, end date of the contract is optional)
 5. Click on the invite button
- **Expected Outcome:**
 - Tenant is successfully invited
 - The property badge changes to **invitation sent**.
 - The tenant receives an e-mail affiliated with the property to create an Keyz account.

Scenario 8: Cancel the Tenant invitation

- **Role Involved:** Property Owner
- **Objective:** Test the cancel invite tenant functionality
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with the badge **invitation sent**
 3. Click on the drop-down menu at the top right of the property and click on the cancel invitation
 4. Click on confirm on the pop up
- **Expected Outcome:**
 - Invitation is successfully canceled
 - The property badge changes to **available**, and all mentions of the tenant disappear from it.
 - The mail that receive the tenant does not work

Scenario 9: See profile

- **Role Involved:** Property Owner & Property Tenant
- **Objective:** Check if the user can see his profile informations
- **Preconditions:**
 - User is logged in with appropriate permissions
- **Test Steps:**
 1. Navigate to settings page
 2. Check the profile informations
- **Expected Outcome:**
 - The informations are well displayed and good

Scenario 10: Change the app language

- **Role Involved:** Property Owner & Property Tenant
- **Objective:** Check if the user can change the language of the app
- **Preconditions:**
 - User is logged in with appropriate permissions

- **Test Steps:**
 1. Navigate to settings page
 2. Click on the other language on the language slider
- **Expected Outcome:**
 - The app change his language
 - The UI is the same, no bugs

Scenario 11: The user can check damages of his property

- **Role Involved:** Property Owner & Property Tenant
- **Objective:** Check if the user can see the damages of his property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
- **Test Steps:**
 1. Navigate to property page and, for the owner click to a property with a tenant
 2. Click on the damages tab
- **Expected Outcome:**
 - The damage are loaded and well displayed
 - There is no bugs

Scenario 12: The tenant can add damages on the property

- **Role Involved:** Property Tenant
- **Objective:** Check if the user can see the damages of his property
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
- **Test Steps:**
 1. Navigate to property page
 2. Click on the damages tab
 3. Click on the reporting a claim button
 4. Fill the form and click confirm
- **Expected Outcome:**
 - The damage does appered on the damages list for both the tenant and the owner
 - There is no bugs

Scenario 13: The owner can end the lease

- **Role Involved:** Property Owner
- **Objective:** End the lease for the
- **Preconditions:**
 - User is logged in with appropriate permissions
 - Property exists in the system
- **Test Steps:**
 1. Navigate to property page
 2. Choose a property with the badge **unavailable**
 3. Click on the drop-down menu at the top right of the property and click on the end lease
 4. Click on confirm on the popup
- **Expected Outcome:**
 - Lease is correctly ended
 - User cannot access to the property
 - The property badge changes to **available**, and all mentions of the tenant disapeard from it.
 - The mail that receive the tenant does not work

Scenario 14: The user can add a room

- **Role Involved:** Property Owner
- **Objective:** Test the creation of a room
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property page
 2. Select a property

3. Click on the property on the properties list with a tenant in it
 4. Click on the inventory button
 5. Click on the add button
 6. Fill the add room form and click confirm
- **Expected Outcome:**
 - The room as been added to the room list
 - The room got the same name and type as filled in the form
 - There is not bug or lags when adding a room
 - Two rooms with the same name cannot be created

Scenario 15: The user can add a furniture

- **Role Involved:** Property Owner
- **Objective:** Test the creation of a room
- **Preconditions:**
 - User is logged in with appropriate permissions
 - A property exists in the system
- **Test Steps:**
 1. Navigate to the property page
 2. Select a property
 3. Click on the property on the properties list with a tenant in it
 4. Click on the inventory button
 5. Click on a room of the list
 6. Click on the add button
 7. Fill the add furniture form and click confirm
- **Expected Outcome:**
 - The furniture as been added to the furniture list of the room
 - The furniture got the same name as filled in the form
 - There is not bug or lags when adding a furniture
 - Two furnitures with the same name cannot be created

3. Success Criteria

The following criteria will be used to determine the success of the beta version.

Criterion	Description	Threshold for Success
Stability	No major crashes	No crash reported by the testers in any of the applications
Stability	No api crashes	No crash that make the api unreachable
Stability	No blocking state	No blocking state reported by the testers
Stability	Api always available	Ensure that the API is at least 90% of the time available
Usability	Understable UI/UX	90% of tester does not get lost on the web and mobile
Usability	Pretty UI	>70% of tester say that the web and mobile application are okay or pretty in terms of design
Usability	distinguishable UI	No elements with bad accessibility on the web and mobile
Usability	Recognisable brand	Unique logo and color that >80% of testers find recognisable
Performance	IA accuracy	the AI give the right answer >70% of time
Performance	IA time frame	the AI responds in less than 90 seconds on average
Performance	Size of inventory report	the inventory report can take up to 5 rooms with 4 elements each
Performance	Apps performance	>85% of testers must not say that they experiences freeze or performances issues with the web and mobile apps
Costs	IA costs	an inventory report must cost less than 2€ on average
Accuracy	Units tests in all of the apps and api	>70% of all the project code lines must be tested
Accuracy	Units tests in all of the apps and api	every major features must have at least 2 tests
Desire	Desire within the testers	>20% of tester should say that if the app was on the market they will use it

4. Known Issues & Limitations

Issue	Description	Impact	Planned Fix? (Yes/No)
iOS Alert Display Bug	On iOS, opening an alert too quickly after closing one grays out the background but the alert doesn't appear	Medium	Yes
Messaging system not working	Messaging system has not been implemented yet	Low	Yes
Notifications on mobile apps	Notifications have not been implemented on mobile apps	Low	Yes

Limitations

- **Tenant Invitation Link Behavior:** If a tenant opens the invitation received by email on their phone, they will be directed to a web page rather than the Keyz application.

5. Conclusion

This Beta Test Plan represents a crucial phase in the development of Keyz, focusing on validating core functionalities essential for property management and tenant interactions. Through structured testing scenarios across both web and mobile platforms, we aim to:

1. **Validate Core Features:** Ensure robust functionality of critical features including property creation, inventory management, and tenant invitation systems.
2. **Cross-Platform Consistency:** Verify seamless user experience across web and mobile interfaces, with special attention to platform-specific interactions.
3. **User Role Verification:** Confirm that both Property Owner and Tenant roles function as intended with appropriate access controls and permissions.
4. **Quality Assurance:** Identify and address potential issues before full release, with documented known limitations to guide future development priorities.

The successful execution of this test plan will provide valuable insights for final refinements and ensure Keyz meets the high standards required for a professional property management solution. Feedback gathered during this beta phase will be instrumental in delivering a polished, user-friendly platform that effectively serves the needs of property owners and tenants alike.