

RIPHAH INTERNATIONAL UNIVERSITY ISLAMABAD  
I-14 CAMPUS



**Project Artifact 02**

**Instructor:** Mr Islam Abasi

**Course:** Software Construction & Development

**Section:** BSSE5-1

**Semester:** 5<sup>th</sup>

NAME	SAP ID
Malik Usama	56340
Rana Anees	47016

## Use Case 1: View City News

**Actor: Resident**

Use Case Section	Comment	
Use Case No	1	
Use Case Name	View City News	
Actor	Resident	
Type	Primary	
Description	This use case allows residents to view daily or weekly city news, updates, and announcements.	
Preconditions	Resident must be logged into the system and have access to the internet.	
Postconditions	Latest city news is displayed successfully to the resident.	
Normal flows	Actor Action:	System Response:
	<b>1.</b> Resident logs into the system.  <b>3.</b> Resident selects the “City News” option.  <b>5.</b> Resident clicks on a specific article.	<b>2.</b> System verifies credentials and loads the dashboard.  <b>4.</b> System retrieves and displays the latest news articles.  <b>6.</b> System shows the full news content with details and date.
Alternative Flow	<b>A4.1.</b> If there is no internet connection, system displays a cached version of previously loaded news.  <b>A4.2.</b> If there are no new updates, system shows “No new news available.”	

## Use Case 2: Report a Problem

Use Case Section	Comment	
Use Case No	2	
Use Case Name	Report a Problem	
Actor	Resident	
Type	Primary	
Description	Allows residents to report issues such as road damage, power outages, or broken streetlights.	
Preconditions	Resident must be logged into the system and have access to the internet.	
Postconditions	The problem report is stored and forwarded to the Admin for review.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Resident logs into the system.  <b>3</b> .Resident selects “Report a Problem.”  <b>5</b> .Resident enters issue details and uploads photo.  <b>7</b> .Resident submits the report.  <b>9</b> .Resident receives confirmation message.	<b>2</b> .System opens the main dashboard.  <b>4</b> .System displays the report form.. <b>6</b> .System validates input and allows adding location.  <b>8</b> .System saves it and generates a unique Report ID. <b>10</b> .System notifies Admin about the new report.
Alternative Flow	<b>A5.1.</b> If GPS or camera access is denied, system requests permission. <b>A6.1.</b> If the user doesn’t fill all required fields, system prompts to complete them. <b>A8.1.</b> If internet is lost, report is stored locally until reconnection.	

### Use Case 3: Get Notifications

Use Case Section	Comment	
Use Case No:	3	
Use Case Name:	Get Notifications	
Actor:	Resident	
Type:	Secondary	
Description:	Allows residents to receive notifications about new news, report updates, or responses from Admins.	
Preconditions:	Resident must be logged into the system and have access to the internet.	
Postconditions:	Notifications are displayed or stored in the user's notification center.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Resident logs into the system.  <b>3</b> .Resident opens the notifications panel.  <b>5</b> .Resident selects a notification.	<b>2</b> .System checks for unread notifications.  <b>4</b> .System displays all recent updates.  <b>6</b> .System displays detailed message.
Alternative Flow	<b>A2.1.</b> If there are no notifications, system shows "No new notifications." <b>A2.2.</b> If push notifications are disabled, system prompts to enable them.	

## Use Case 4: Search Items

Use Case Section	Comment	
Use Case No:	4	
Use Case Name:	Search Items	
Actor:	Resident	
Type:	Primary	
Description:	Enables residents to search for products, services, or businesses within the city.	
Preconditions:	Resident must be logged into the system and have access to the internet.	
Postconditions:	The system displays matching search results.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Resident logs into the system.  <b>3</b> .Resident selects “Search.”  <b>5</b> .Resident enters name, category, or location.  <b>7</b> .Resident views results.	<b>2</b> .System displays dashboard.  <b>4</b> .System displays search filters.  <b>6</b> .System retrieves matching data .  <b>8</b> .System displays details of selected item.
Alternative Flow	<b>A6.1.</b> If no results match, system shows “No items found.” <b>A6.2.</b> If server is busy, system retries or shows cached results.	

## Use Case 5: Give Feedback

Use Case Section	Comment	
Use Case No:	5	
Use Case Name:	Give Feedback	
Actor:	Resident	
Type:	Secondary	
Description:	Allows residents to provide feedback or rate businesses and city services.	
Preconditions:	Resident must be logged into the system and have access to the internet.	
Postconditions:	Feedback is successfully submitted and saved in the database.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Resident logs into the system.  <b>3</b> .Resident selects a business or service.  <b>5</b> .Resident enters comments and rating.  <b>7</b> .Resident submits feedback.	<b>2</b> .System loads the feedback section.  <b>4</b> .System displays feedback form.  <b>6</b> .System validates input.  <b>8</b> .System saves feedback and confirms submission.
Alternative Flow	<b>A6.1.</b> If feedback is empty, system prompts user to write comments. <b>A8.1.</b> If server error occurs, system retries later.	

## Use Case 6: Get Suggestions

Use Case Section	Comment	
Use Case No:	6	
Use Case Name:	Get Suggestions	
Actor:	Resident	
Type:	Secondary	
Description:	Provides personalized suggestions based on user's interests and location.	
Preconditions:	Resident must be logged into the system and have access to the internet.	
Postconditions:	Relevant suggestions are displayed on the user dashboard.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Resident logs into the system.  <b>3</b> .Resident opens "Suggestions."	<b>2</b> .System analyzes user data and location.  <b>4</b> .System displays personalized recommendations.
Alternative Flow	<b>A2.1.</b> If location permission denied, system prompts to enable it. <b>A4.1.</b> If no suggestions available, system displays "No recommendations yet."	

## Use Case 1: Add Business Details

**Actor: Business Owner**

Use Case Section	Comment	
Use Case No:	1	
Use Case Name:	Add Business Details	
Actor:	Business Owner	
Type:	Primary	
Description:	This use case allows the business owner to add details about their business, such as name, address, contact info, and description, to make it visible in the Link City System.	
Preconditions:	Business owner must be logged into the system and have access to the internet.	
Postconditions:	The business details are stored and submitted to the Admin for approval.	
Normal flows	Actor Action:	System Response:
	<p>1 .Business owner logs into the system.</p> <p>3 .Business owner selects “Add Business.”</p> <p>5 .Owner enters business name, address, contact details, and description.</p> <p>7 .Owner submits the form.</p> <p>9 .System confirms submission.</p>	<p>2 .System loads the dashboard.</p> <p>4 .System displays the business registration form.</p> <p>6 .System validates all fields.</p> <p>8 .System saves the business details and sends them to Admin for review.</p> <p>10 .Admin receives notification for approval.</p>
Alternative Flow	<p>A6.1. If any required field is missing, the system displays an error message.</p> <p>A6.2. If the business already exists, the system alerts the owner about duplication.</p> <p>A8.1. If internet connection fails, data is saved locally until connection is restored.</p>	



## Use Case 2: Add Product Details

Use Case Section	Comment	
Use Case No:	2	
Use Case Name:	Add Product Details	
Actor:	Business Owner	
Type:	Primary	
Description:	This use case allows business owners to add or update products offered by their business, including product name, price, description, and availability.	
Preconditions:	Business owner must be logged into the system, and their business must be approved by the Admin.	
Postconditions:	Product details are successfully saved and displayed under the business profile.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Business owner logs into the system.  <b>3</b> .Owner selects their approved business.  <b>5</b> .Owner clicks “Add Product.”  <b>7</b> .Owner enters product name, price, description, and uploads image.  <b>9</b> .Owner receives confirmation message.	<b>2</b> .System opens the dashboard.  <b>4</b> .System displays options to manage products.  <b>6</b> .System opens product form.  <b>8</b> .System validates data and saves the product.  <b>10</b> .System updates the business product list.
Alternative Flow	<b>A3.1</b> .If the business is not yet approved, system shows message “Cannot add products until business is approved.”  <b>A8.1</b> . If any mandatory detail is missing, the system prompts to fill it.  <b>A7.1</b> . If image upload fails, the system allows retry.	

### Use Case 3: Respond to Customer Feedback

Use Case Section	Comment	
Use Case No:	3	
Use Case Name:	Respond to Customer Feedback	
Actor:	Business Owner	
Type:	Secondary	
Description:	Allows business owners to read and respond to feedback or comments left by customers.	
Preconditions:	Business owner must be logged into the system and the business must have at least one feedback entry.	
Postconditions:	The response is recorded and visible to the customer and Admin.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Business owner logs into the system.  <b>3</b> .Owner selects “Customer Feedback.”  <b>5</b> .Owner selects a feedback item.  <b>7</b> .Owner writes a reply message.	<b>2</b> .System shows feedback notifications.  <b>4</b> .System displays list of feedback entries.  <b>6</b> .System displays detailed view with comment and rating.  <b>8</b> .System saves the reply and notifies the customer.
Alternative Flow	<b>A4.1.</b> If there is no feedback, system displays “No feedback available.” <b>A8.1.</b> If internet connection fails, reply is saved locally for later upload.	

## Use Case 4: View Business Performance

Use Case Section	Comment	
Use Case No:	4	
Use Case Name:	View Business Performance	
Actor:	Business Owner	
Type:	Secondary	
Description:	This use case allows the business owner to view business insights such as traffic, ratings, and customer engagement.	
Preconditions:	Business owner must be logged into the system and the business must be active.	
Postconditions:	System displays business performance data and analytics.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Business owner logs into the system.  <b>3</b> .Owner selects “View Business Performance.”  <b>5</b> .Owner views reports with reviews and feedback..	<b>2</b> .System loads business dashboard.  <b>4</b> .System retrieves business analytics from database.  <b>6</b> .System displays graphs, ratings, and recent activity.
Alternative Flow	<b>A4.1.</b> If analytics data is unavailable, system shows “No data found.” <b>A4.2.</b> If internet is slow, system displays partial data with refresh option.	

## Use Case 1: Review Reports

**Actor: Admin (City Official)**

Use Case Section	Comment	
Use Case No:	1	
Use Case Name:	Review Reports	
Actor:	Admin (City Official)	
Type:	Primary	
Description:	This use case allows the Admin to review issues submitted by residents and decide whether to approve or reject them.	
Preconditions:	Admin must be logged into the system and have access to the internet.	
Postconditions:	The report is reviewed and either approved or rejected by the Admin.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Admin logs into the system.  <b>3</b> .Admin selects “Review Reports.”  <b>5</b> .Admin clicks on a specific report to view details.  <b>7</b> .Admin reviews the report and selects an action.	<b>2</b> .System displays the Admin dashboard.  <b>4</b> .System lists all pending reports submitted by residents.  <b>6</b> .System displays report description, photo, and location.  <b>8</b> .System updates report status and notifies the respective resident.
Alternative Flow	<b>A6.1.</b> If the report contains incomplete information, system prompts Admin to request more details.  <b>A7.1.</b> If internet is lost, review process is paused until connection is restored.	

## Use Case 2: Manage Records

Use Case Section	Comment	
Use Case No:	2	
Use Case Name:	Manage Records	
Actor:	Admin (City Official)	
Type:	Primary	
Description:	This use case enables the Admin to manage all user accounts, business listings, and system categories to maintain proper structure.	
Preconditions:	Admin must be logged into the system and connected to the database.	
Postconditions:	Selected records are successfully updated or modified as per the Admin's actions.	
Normal flows	Actor Action:	System Response:
	1 .Admin logs into the system.	2 .System displays management dashboard.
	3 .Admin selects "Manage Records."	4 .System shows available record types (Users, Businesses, Categories).
	5 .Admin selects a record type .	6 .System loads the relevant records.
	7 .Admin edits, adds, or removes data.	8 .System validates and saves changes.
Alternative Flow	<b>A6.2.</b> If database is locked or unavailable, system alerts Admin and retries later. <b>A8.1.</b> If invalid data is entered, system displays an error message.	

### Use Case 3: Create Reports

Use Case Section	Comment	
Use Case No:	3	
Use Case Name:	Create Reports	
Actor:	Admin (City Official)	
Type:	Primary	
Description:	This use case allows the Admin to generate summary reports about system activity, business performance, or resident issues.	
Preconditions:	Admin must be logged into the system and have access to all relevant data.	
Postconditions:	The requested report (daily or weekly) is successfully generated and saved.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Admin logs into the system.  <b>3</b> .Admin selects “Create Reports.”  <b>5</b> .Admin selects “Daily” or “Weekly” (<extend>).  <b>7</b> .Admin confirms report creation.	<b>2</b> .System displays reporting options.  <b>4</b> .System prompts for type and date range.  <b>6</b> .System retrieves relevant data.  <b>8</b> .System generates report and saves it in database.
Alternative Flow	<b>A6.1.</b> If data is missing, system notifies Admin and skips incomplete sections. <b>A8.1.</b> If report generation fails, system allows retry.	

## Use Case 4: Manage Information

Use Case Section	Comment	
Use Case No:	4	
Use Case Name:	Manage Information	
Actor:	Admin (City Official)	
Type:	Secondary	
Description:	Allows the Admin to edit or delete incorrect or harmful information from the system.	
Preconditions:	Admin must be logged into the system and have access permissions for modification.	
Postconditions:	Information is updated or deleted successfully in the system.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Admin logs into the system.  <b>3</b> .Admin selects “Manage Information.”  <b>5</b> .Admin chooses a record to edit or delete (<extend>).  <b>7</b> .System confirms the update or deletion.	<b>2</b> .System displays data management options.  <b>4</b> .System loads available data records.  <b>6</b> .System executes the requested action.  <b>8</b> .System logs the action for transparency..
Alternative Flow	<b>A6.1.</b> If Admin tries to delete restricted data, system denies access. <b>A6.2.</b> If an edit fails validation, system requests correct data.	

## Use Case 5: Keep Records

Use Case Section	Comment	
Use Case No:	5	
Use Case Name:	Keep Records	
Actor:	Admin (City Official)	
Type:	Secondary	
Description:	This use case allows the Admin to maintain logs of all system actions, including user activity, reports, and updates for transparency and auditing.	
Preconditions:	Admin must be logged into the system and have access to the data storage.	
Postconditions:	Records are securely stored and can be retrieved when required.	
Normal flows	Actor Action:	System Response:
	<b>1</b> .Admin logs into the system.  <b>3</b> .Admin selects “Keep Records.”  <b>5</b> .Admin reviews, filters, or exports data.	<b>2</b> .System opens record management dashboard.  <b>4</b> .System displays log options (User logs, Business logs, Report logs).  <b>6</b> .System updates logs and ensures they are securely backed up.
Alternative Flow	<b>A6.1.</b> If storage is full, system alerts Admin to free space. <b>A6.2.</b> If backup fails, system retries automatically.	