Release Notes & User Manual
Version 1.0
DEC 12, 2025

SyncBridge

Group 6
CHEN Tianyu 1155210962
GUO Mingkang 1155211019
HUA Zifan 1155211089
Li Molin 1155191354
Qi Zihan 1155191644

Submitted in partial fulfillment
Of the requirements of
CSCI 3100 Software Engineering

1. Introduction

1.1 About SyncBridge

SyncBridge is a web-based collaboration platform that connects clients and developers around software requirements. It provides a main requirement form (mainform) for initial requirement submission, a subform negotiation mechanism to propose and accept changes, structured functional and non-functional requirement items, threaded messaging blocks (general / function / nonfunction), real-time communication, and automatic email reminders for urgent and normal discussions.

This manual explains how to access and run the SyncBridge application, how to use SyncBridge as a client or developer, how to work with forms, subforms, functions, nonfunctions, messages, files, and statuses, and how to understand expected system behaviour, assumptions, and limitations.

1.2 Intended Audience

This manual is intended for clients who submit and manage requirements, and developers who accept, negotiate, and implement requirements.

2. System Access and Prerequisites

2.1 Browser Requirements

A modern browser is required. It is recommended to use the latest version of Google Chrome, Microsoft Edge, or Firefox. JavaScript and cookies must be enabled. A screen resolution of at least 1280×720 is recommended. The layout is responsive and supports smaller screens, but more scrolling may be needed.

2.2 Network Requirements

A stable internet connection is required. The client must be able to reach the SyncBridge backend and WebSocket endpoints over standard HTTPS/WSS ports.

2.3 Register & Login

New users can register by entering their name, email, and password, and by choosing a user type (client or developer). After registration, the account is created and the user is logged in automatically, typically as a client by default.

Existing users log in using their name, email, and password. On successful login, the user is taken to the dashboard or form list, and the session is maintained for the current browser tab.

A Logout button or link in the navigation clears the session and returns the user to the login page. After logging out, protected pages are no longer accessible until the user logs in again.

4. Using SyncBridge as a Client

4.1 Dashboard

As a client, the dashboard shows the user's form list. For each form, the dashboard displays the title, status, last updated time, and badges such as "Subform", "Urgent", or "Changes pending".

4.2 Creating a Requirement

To create a new requirement, the user goes to Forms or the Dashboard and selects "Create New Form". The user then fills in the title, description, budget, and expected time, and can optionally add functions and nonfunctions. The form can be saved, usually with status preview. When the form is ready, the user submits it to make it available. After submission, developers can see it in the list of available forms, and the client can still edit it if the status and rules allow.

4.3 Viewing a Form

To view a form, the user goes to Forms, selects a form, and opens the Form Detail page. The Form Detail page shows the mainform fields and status, indicates whether a subform exists, and displays the lists of functions and nonfunctions.

4.4 Editing a Form

Editing a form is allowed when the status is preview or available, and when the backend permits changes. The user opens the Form Detail page and uses edit buttons on the mainform fields and on the functions or nonfunctions. After saving, the UI updates immediately. If editing is not allowed, the edit buttons are not shown or an error message appears.

4.5 Status

The preview status indicates a draft that is usually editable and not yet a full task. The available status indicates that developers can accept the form or create a subform. The processing status indicates that the form has been accepted and is in progress; messaging remains open. The rewrite status indicates that a subform exists, differences are highlighted, and the form is in negotiation. The end status indicates that the form is finished and read-only, but the history is still viewable. The error status indicates a problem or dispute and is used for further communication.

4.6 Subform Negotiation

When a subform exists, the Form Detail page shows the mainform and subform side by side or with highlighted differences. Changed fields are visually marked, for example with an "is_changed" flag. The client can review changes to functions, nonfunctions, budget, and other fields, and can discuss these changes in message blocks.

If the client agrees with the proposed changes, the client can merge or accept the subform. If the client disagrees, the client can reject or delete the subform, if permitted. On merge, the subform overwrites the mainform, the subform is removed, and the status usually changes to processing. On rejection, the subform is deleted and the mainform returns to its previous state.

4.7 Messaging (Client)

For a given form, the client opens the Message Room to communicate with the developer. The client can switch between blocks such as General, Function, and Nonfunction. The client types a message,

optionally attaches files, and sends it. Messages appear instantly via WebSocket, and the developer sees them in real time.

4.8 File Upload

In the message input area, the client can attach a file by selecting “Attach File”, choosing a file of up to 10 MB, and sending the message. The message and file appear in the thread, and the other party can preview or download the attachment. Images are shown with an inline preview. Text and code files are displayed in a formatted way. Other binary files are shown with an icon, file name, and download link.

5. Using SyncBridge as a Developer

5.1 Developer View

As a developer, the main view shows lists of available forms (unassigned), processing forms (assigned to the developer), and end forms (completed by the developer). Each entry displays the title, status, and badges such as “Subform”, “Urgent”, or “Negotiations”.

5.2 Accepting a Form

To accept a form, the developer opens the Available Forms list, selects a form, and reviews the mainform, functions, nonfunctions, and messages. The developer then clicks “Accept”. The status usually changes from available to processing, and the form is assigned to the developer.

5.3 Proposing Changes

If changes are needed, the developer opens the Form Detail page and chooses “Propose Changes” or “Create Subform”. In the subform, the developer can edit main fields such as title, description, budget, and expected time, and can add, edit, or delete functions and nonfunctions. Changes are marked as changed. After saving, the status becomes rewrite. The client can then review, discuss, merge, or reject the subform.

5.4 Editing Functions / Nonfunctions

Editing functions and nonfunctions is usually allowed when the status is processing or rewrite, and when the rules permit. The developer can create, edit, or delete function and nonfunction items. Changes appear in the UI and may be flagged with an “is_changed” indicator.

5.5 Negotiation and Merge

When the status is rewrite, the developer compares the mainform and subform, with differences highlighted. General, function, and nonfunctional requirement messages are used to justify and discuss changes. When both sides agree, the developer or client can trigger a merge, if allowed. On merge, the subform overwrites the mainform, the subform is removed, and the status changes to processing. If negotiations fail, the subform can be deleted or rejected, if permitted. In that case, the mainform returns to its previous state, and the developer may continue work or set the status to error, depending on the context.

5.6 Completing a Form

When the work is finished, the developer opens the Form Detail page and selects “Finish” or “Complete”. The status changes from processing to end. The form becomes read-only, but both sides can still view messages and attachments.

6. Messaging, WebSocket, and Email Behaviour

6.1 Real-Time Messaging

SyncBridge uses a WebSocket connection to deliver messages in real time. As long as the browser tab is open and the network is connected, new messages appear automatically without a page refresh. If the WebSocket connection is lost, the system attempts to reconnect automatically. The user may see notifications if the connection is unstable.

6.2 Message History and Pagination

Older messages can be loaded by scrolling up in the message list or by using a “Load more” button, depending on the UI. The system fetches messages in pages to keep performance acceptable for long discussions.

6.3 Email Reminders and Block Urgency

Each block, or discussion thread, has an urgency setting. For an urgent block, if there is no reply within 5 minutes, the system sends an email reminder to the other party. For a normal block, if there is no reply within 48 hours, the system sends an email reminder. When the status of a form is updated, the system sends an email to both the client and the developer.

Email reminders are sent by the backend only; there is no manual email “send” button in the UI. Each idle period triggers at most one reminder, until a new message is posted and the timer is reset.

7. Known Assumptions and Limitations

7.1 Assumptions

The system assumes that users have stable network connectivity and that they work primarily from desktop or laptop browsers. Each user has exactly one license, either client or developer, at a time. For each mainform, there is only one active subform.

7.2 Limitations in Version 1.0

In Version 1.0, there is no admin role; only client and developer roles are available. There is no multi-version history. Once a subform is merged or rejected, it is not stored as a separate historical version. File size for uploads is limited to 10 MB; larger files must be transferred outside the system. Email content is simple and focuses on block inactivity (urgent and normal), status changes, and subform events.

Appendix

Main Concepts

This section explains the key terms used in the system.

Mainform
The main requirement form created by a client. It includes title, description, budget, expected time, and lists of functions and nonfunctions. Each mainform has a status showing its progress.

Subform
A temporary copy of a mainform used for negotiation. Created by the client/developer when proposing changes. Only one active subform per mainform. Differences from the mainform are highlighted. When both sides agree, the subform can be merged into the mainform.

Functions (Functional Requirements)
Items that describe what the system should do. They belong to a mainform or subform and typically have a name and description.

NonFunctions (Non-Functional Requirements)
Items that describe quality attributes (for example performance, reliability). They also belong to a form and typically have a name, level, and importance.

Blocks and Messages
A Block is a discussion thread. Types:

- General: about the whole form
- Function: about one function
- Nonfunction: about one non-functional requirement

A Message is a single post in a block, with text, optional attachment, timestamp, and author. The block's urgency controls email reminders.

Status

- preview: draft
- available: visible to developers, can be accepted
- processing: accepted by a developer, in progress
- rewrite: in negotiation, a subform exists
- end: finished, read-only
- error: error or dispute

Different statuses allow different actions in the UI.

License / Role

- client: requirement owner
- developer: implementer

Your role decides which pages and actions you have.

SyncBridge — System Release Notes

Version: 0.9 (Course Demo Release)
Release Date: 22 Dec 2025
Release Type: Course Demo Snapshot

Overview

This release is a **course demo snapshot** of the SyncBridge system. The system comprises a backend core service and a frontend prototype. At this stage, **full end-to-end integration is not assumed**. The release focuses on demonstrating the **system design, core backend logic, and intended behaviors** rather than a production-ready deployment.

Backend Status

The backend is **substantially implemented and stable** for the demo. It includes JWT-based authentication, license activation and role-based access control (client/developer), requirement submission and lifecycle management (preview, available, processing, rewrite, error, end), subform-based negotiation, block-based messaging with WebSocket real-time updates, file upload with a 10MB limit, email reminders for urgent and normal discussions, and audit logging for key operations.
Some capabilities remain partial or deferred, including event-driven notifications beyond reminders, audit log querying/export, file preview support, and automated testing/monitoring.

Frontend Status

The frontend in this release is a **prototype**. An earlier implementation was developed independently and does not strictly follow the finalized backend API definitions. As a result, some demo behaviors may rely on mocked data or partial integration. A **frontend rewrite** is planned to fully align with the backend APIs, targeted for completion **by 24 Dec 2025** for the final demo video.

Integration and Limitations

End-to-end frontend–backend integration is **not assumed**. Some behaviors are demonstrated via backend-level verification or frontend mocks. The current release excludes admin roles, password recovery, user-configurable notification preferences, and formal performance/accessibility testing.

Documentation Scope

This release note, together with the User Manual, documents the **intended and demo-supported behavior** at the time of submission. Any discrepancies with live demo output are due to the current integration status and are explicitly acknowledged.

GitHub Repository

https://github.com/normalman743/SyncBridge