

Booking Code Converter

Assignment
Version History
 Environment
Platform
Region
Back office related
 Timeline
Feature Description
User Story
Error Handling
Details (SPEC)

Architecture / Components
Flows
1. Code2Code – in-app (text input)
2. OCR2Code – in-app (share image)
2. OCR2Code – Telegram bot
3. Prompt2Code – app widget or Telegram (Phase 2)

Use Case
BO Page Design
Copywriting
GA events
 Follow-up
Metrics
DA requirements
Design assets & inspiration
Others

Assignment

Epic ticket	 FOOTBALL-9161: Booking Code Converter BACKLOG
Document status	DRAFT 1ST VERSION ...
Document owner	@Dino Yeh
UED	@Ten Liu
Developers	BE: @Gary Chen @Danny Lin Android: @David Huang WAP: @Paul Yang IOS: @Reed Hsin
QA	
DA	@Richard Cheng
Figma	 https://www.figma.com/design/SvcTIADMZ7gUPIa7nN2hT1/Code-Converter?node-id=24353-69&t=YPZwA72g psUABU2r-11 連結 Figma 帳戶
Api Doc	

System Design	BE: Android: WAP: IOS:
----------------------	---------------------------------

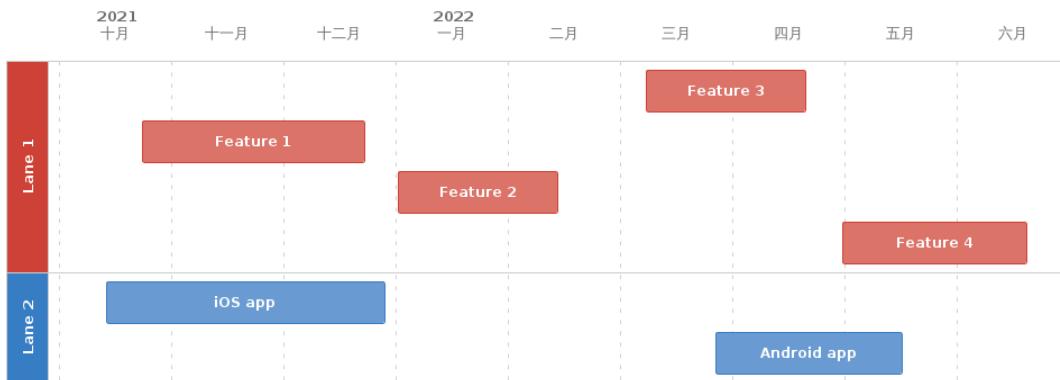
Version History

Date	Memo

Environment

Platform	Region	Back office related
<input checked="" type="checkbox"/> BE <input checked="" type="checkbox"/> WAP <input type="checkbox"/> FE for BO page <input checked="" type="checkbox"/> Android <input checked="" type="checkbox"/> iOS	<input checked="" type="checkbox"/> NG <input checked="" type="checkbox"/> GH	<input type="checkbox"/> BO Page <input checked="" type="checkbox"/> BO Config

Timeline



Feature Description

- Allow users to easily convert competitor booking codes or bet slip screenshots into Fcom booking codes and bet directly on Fcom. In a later phase, allow users to generate betting combinations via natural language prompts.

Phase 1 Scope

- Code2Code (Competitor → Fcom booking code)**
 - User inputs a competitor booking code in Fcom app (or via Telegram bot).
 - Backend calls competitor public “load code” APIs, retrieves the bet composition (events, markets, outcomes), maps them to Fcom’s offers, and generates an Fcom booking code.
 - The generated Fcom booking code is automatically loaded into Fcom betsliip.

Phase 2 Scope

- **OCR2Code (Image → Fcom booking code)**

- User shares a bet screenshot (from competitor / social media) to Fcom app or Telegram bot.
- Backend uses OCR + parsing to extract selections, maps them to Fcom offers, generates an Fcom booking code, and loads it into betslip (or replies in TG).

Phase 3 Scope

- **Prompt2Code (Prompt → Fcom booking code)**

- User sends a natural language prompt (e.g. “find me 3 games starting today with the highest win probability”) via Fcom app widget or Telegram bot.
- Backend uses AI to interpret the prompt, fetches suitable selections from Fcom DB, constructs a booking code, and returns it.

Platforms

- Android/IOS/WAP
- Telegram bot (existing)
- Shared backend service: **Booking Code Generator**

Key value

- Reduce friction for users migrating from other bookies or copying social media bet slips.
- Capture external/social betting traffic into Fcom.
- Provide “smart bet suggestions” (Phase 2) using existing Data + AI.

User Story

Story Name	As a...	I want to...	So that...
Code2Code (within Fcom app)	As a user who sees an interesting bet from another bookie	I want to paste the competitor's booking code into Fcom	so that I can quickly place the same bet on Fcom without manually rebuilding it.
OCR2Code (Fcom app – share image)	As a user browsing social media / chat groups	I want to share a bet screenshot directly to Fcom	so that Fcom can read it, convert it, and let me bet the same combination easily.
OCR2Code (Telegram bot)	As a user who uses Telegram more than the app,	I want to forward a bet screenshot to the Fcom Telegram bot	so that it can convert it into a Fcom booking code or selections I can reuse.
Prompt2Code (Phase 2)	As a user who doesn't have a specific code, but has an idea,	I want to type a simple request (e.g. “3 safe picks today, highest win probability”)	so that Fcom can generate a booking code for me based on my preferences.

Async / long-running flow	As a user	I want to be able to leave the screen while the system is converting my image or prompt	so that I don't feel blocked by long AI/OCR processing times and can continue using the app.
---------------------------	-----------	---	--

Error Handling

- **Invalid competitor code**

- Message: “We couldn’t find this booking code on [BookieName]. Please check and try again.”

- **OCR cannot parse image**

- Message: “We couldn’t recognize a valid bet from this image. Please try another screenshot or a clearer image.”

- **Partial mapping**

- Message: “We converted some of the selections, but a few are not available on Fcom.”
- Show list of missing events/markets if possible.

- **Timeouts / internal errors**

- Message: “The conversion is taking longer than expected. Please try again later.”
- Log for monitoring and retry options.

Details (SPEC)

Architecture / Components

- **Frontend:**

- Fcom app “Load Code” widget
- Chat-like interface (for async flow & history)
- OS-level share entry (Android/iOS)
- Telegram bot (existing, reusing same backend)

- **Backend:**

- **Sport Booking Code Generator Service**

- Input types:
 - type = COMPETITOR_CODE (raw text code + source bookie)
 - type = IMAGE (image URL or file reference)
 - type = PROMPT (free text, later phase)

- Responsibilities:

1. Parse input (code / image / prompt).
2. Normalize to internal selection structure:
 - event id / name
 - market id / name
 - outcome id / name
 - odds (if needed for mapping / validation)

3. Map external selections to Fcom offers (DB lookup + mapping rules).
4. Build Fcom booking code.
5. Return:
 - booking code
 - normalized selection list
 - error / partial match info if any.

- **Competitor API clients**

- For each supported competitor:
 - load_code endpoint config (URL, auth, rate limit, etc.)
 - Response mapping to internal schema

- **Mapping layer**

- Mapping logic between competitor events/markets/outcomes and Fcom's internal IDs:
 - by event date/time + team names + league
 - by market type name / code
 - by outcome label (e.g. "Home Win", "Over 2.5")

- **AI / OCR layer**

- Accepts image
- Extracts:
 - bookmaker name (if possible)
 - list of events / teams / markets / odds
- Returns a machine-readable structure for the mapping layer.

- **Job / async processing**

- Long-running tasks (OCR / prompt) should be queued:
 - Request ID generated
 - Status: PENDING → PROCESSING → SUCCESS / FAILED / PARTIAL
- Frontend can poll or receive push notification / in-app message when done.

Flows

Phase	Inputs	Flow:	scenarios	Memo
1. Code2Code – in-app (text input) <u>Mockup</u>	<ul style="list-style-type: none"> • User manually enters competitor booking code into Load Code widget. • User selects competitor bookie from dropdown (e.g. Bet9ja, BetKing, etc.). 	<ol style="list-style-type: none"> 1. User opens Load Code widget on home. 2. User selects Bookie from dropdown. 3. User inputs booking code text and taps Load Code. 4. App sends request to Booking Code Generator: 	<ol style="list-style-type: none"> 1. Home page load code widget 2. Empty mini betslip <ol style="list-style-type: none"> a. Refine empty betslip. 3. Code center load code tab 	<ul style="list-style-type: none"> • Invalid competitor code / API returns "not found" • Partial mapping only (some events not available on Fcom) • Competitor API down / timeout monitor @Gary Chen <p>UI should show clear messages, e.g.:</p>

		<p>5. Backend calls competitor's load_code API, gets selections.</p> <p>6. Backend maps those selections to Fcom offers via DB / mapping layer.</p> <p>7. Backend generates Fcom booking code (e.g. AF1234) and returns response</p> <p>8. App:</p> <ul style="list-style-type: none"> ◦ Shows success message: "Converted to Fcom code AF1234" ◦ Automatically opens betslip with selections loaded. 		<ul style="list-style-type: none"> • "Code not found on [BookieName]. Please check and try again." • "We could only convert 3 out of 5 selections. Missing events: ..."
2. OCR2Code – in-app (share image)	<ul style="list-style-type: none"> • User taps OS share on a screenshot. • Chooses Fcom app from share options. 	<p>1. User sees bet screenshot on social / chat → taps Share.</p> <p>2. In system share sheet, user selects Fcom.</p> <p>3. Fcom app opens a "Code Converter" chat UI (or dedicated screen) showing:</p> <ul style="list-style-type: none"> ◦ The shared image ◦ A message "We're converting your screenshot into a booking code..." <p>4. App sends request to backend:</p> <p>5. Backend pushes job to OCR/AI queue:</p>		<ul style="list-style-type: none"> • User should be able to leave the screen, continue other things. • When result is ready: <ul style="list-style-type: none"> ◦ Notification / in-app message ◦ Chat history retains input image + output code

	<p>a. OCR extracts text → events / markets / odds.</p> <p>b. Mapping layer resolves them to Fcom offers.</p> <p>c. Booking code generated.</p> <p>6. Once done, backend sends result:</p> <ul style="list-style-type: none"> ◦ SUCCESS: Fcom booking code + selection list ◦ PARTIAL: some selections missing ◦ FAILED: cannot parse / no matches <p>7. App:</p> <ul style="list-style-type: none"> ◦ In chat UI, shows a message: <ul style="list-style-type: none"> ▪ “Here’s your converted code: AF1234. Tap to bet.” ◦ When user taps, open betslip with selections.
2. OCR2Code – Telegram bot	<p>1. User forwards or uploads a bet screenshot to Fcom TG Bot.</p> <p>2. Bot receives image and sends to backend with:</p> <p>3. Backend runs same OCR + mapping + booking code generator.</p> <p>4. Bot replies in chat:</p> <ul style="list-style-type: none"> ◦ On success:

	<ul style="list-style-type: none"> ▪ “Converted booking code: AF1234” ▪ Optionally list selections. ◦ On partial/failed: explain what went wrong. 	
3. Prompt2Code – app widget or Telegram (Phase 2)	<ol style="list-style-type: none"> 1. User opens Code Converter chat (app) or TG Bot and types a prompt, e.g.: <ol style="list-style-type: none"> a. “Find 3 matches starting today with the highest win probability.” 2. Client sends request 3. Backend: <ol style="list-style-type: none"> a. Uses AI model to parse intent & constraints: <ul style="list-style-type: none"> ▪ number of events: 3 ▪ date: today ▪ criteria: highest win probability b. Queries Fcom DB / recommendation engine for suitable selections. c. Constructs booking code. 4. Returns result → booking code + explanation. 5. UI shows: <ul style="list-style-type: none"> ◦ “Here is a code based on your request: AF5678” ◦ List events & markets. 	

Rank	Bookies	Country	Author
1	Sporty	NG, GH	
2	MSport	NG/GH/UG/ZM	@Danny Lin
3	Bet9ja	--	@Gary Chen
4	Bangbet	KE, UG, NG, GH, TZ	
5.	BetKing	NG	
6	1xBet	<div style="display: flex; justify-content: space-around;"> 1xbet - Nigeria 1xbet - Global 1xbet - Cameroon </div> <div style="display: flex; justify-content: space-around;"> 1xbet - Ghana 1xbet - Kenya 1xbet - Uganda </div>	
7	Betway	Global	
8	LivScore Bet	 LiveScore Bet UK Online Sports Betting & Odds Site	
9	Stack	Global	
10	BetPawa		
11	Bet365	Global	

Use Case



1. In-app text code conversion

- User sees a booking code on a competitor site.
- Opens Fcom app → Load Code widget → selects competitor → inputs code → taps Load → Fcom converts and loads betslip.

2. In-app image share conversion

- User sees a bet screenshot on Instagram / WhatsApp.
- Taps Share → selects Fcom → app opens chat view and uploads image → system converts → returns Fcom booking code → user taps to bet.

3. Telegram screenshot conversion

- User forwards a bet screenshot to Fcom TG bot.
- Bot replies with Fcom booking code and selection list.

4. Prompt-based booking code (Phase 2)

- User opens Code Converter chat in Fcom app.

- Types: “3 matches today, highest win probability, total odds around 3.0.”
- System returns a booking code built from DB with matching criteria.

BO Page Design

Copywriting

GA events

Event name:

trigger	comment

Follow-up

Task	Owner	Status	Action items
		IN PROGRESS / COMPLETE	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Metrics

✓ North Star Metric (Phase 1): Converter-attributed Betslip Loads

Definition: The number of sessions/users who successfully convert a competitor code and **load the resulting selections into the betsip**.

Inclusion criteria (attribution):

- The conversion result is **Success** (or **Partial**, if Phase 1 counts partial loads as valid) and the betsip is loaded.
- The betsip load event must carry `source = "code_converter"` and a unique `conversion_request_id` (or equivalent job/request identifier) to ensure end-to-end traceability.

Notes:

- This metric focuses on the Phase 1 core value: *reducing friction by turning external codes into an actionable betsip on Fcom.*
- It does **not** require a bet placement after the betsip is loaded.

DA requirements

F

Design assets & inspiration

▼ Designs of other products

- **Home – Load Code widget (existing):**
 - **Case 1 (text input):**
 - Bookie dropdown (required for competitor mapping)
 - Text field for code
 - “Load code” CTA
 - **Case 3 (prompt) (Phase 2):**
 - Additional mode / toggle: “Ask AI to build a code”
 - Opens chat view or expands into prompt input.
- **Chat-style interface (recommended for OCR & Prompt):**
 - Pros:
 - Handles slow AI / OCR naturally (async).
 - Clear history: user can see previous images/requests & results.
 - Easy to extend to future features (e.g. explanations / tips).
- **Notifications:**
 - If processing takes > X seconds, show:
 - “We’re still working on it, you can continue using the app. We’ll notify you when it’s ready.”

Others

1