

# 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)  
OCR2Code – in-app (share image)  
OCR2Code – Telegram bot  
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	 <a href="https://www.figma.com/design/SvcTIADMZ7gUPIa7nN2hT1/Code-Converter?node-id=24353-69&amp;t=YPZwA72g">https://www.figma.com/design/SvcTIADMZ7gUPIa7nN2hT1/Code-Converter?node-id=24353-69&amp;t=YPZwA72g</a> psUABU2r-11 連結 Figma 帳戶
Api Doc	

<b>System Design</b>	BE: Android: WAP: IOS:
----------------------	---------------------------------

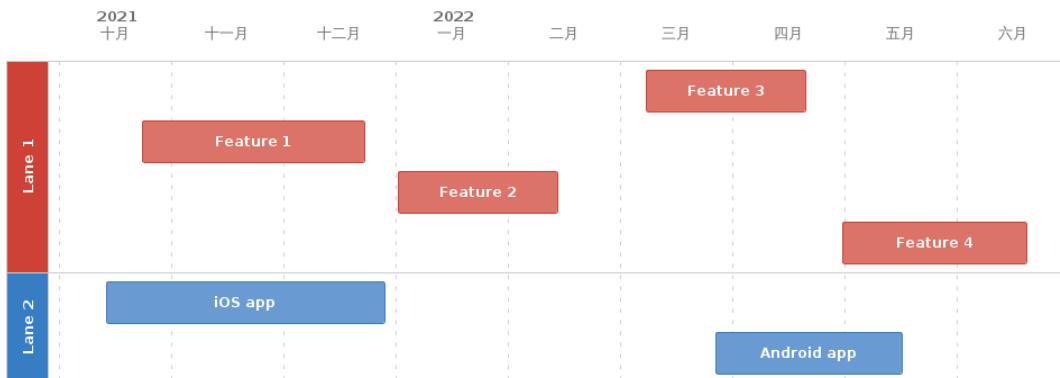
## Version History

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"> <li>User manually enters competitor booking code into Load Code widget.</li> <li>User selects competitor bookie from dropdown (e.g. Bet9ja, BetKing, etc.).</li> </ul>	<ol style="list-style-type: none"> <li>User opens <b>Load Code</b> widget on home.</li> <li>User selects <b>Bookie</b> from dropdown.</li> <li>User inputs <b>booking code</b> and taps <b>Load Code</b>.</li> </ol>	<ol style="list-style-type: none"> <li>Home page load code widget</li> <li>Empty mini betslip           <ol style="list-style-type: none"> <li>Refine empty betslip.</li> </ol> </li> <li>Code center load code tab</li> </ol>	<ul style="list-style-type: none"> <li>Invalid competitor code / API returns "not found"</li> <li>Partial mapping only (some events not available on Fcom)</li> <li>Competitor API down / timeout monitor</li> </ul>

		<p>4. App sends request to <b>Booking Code Generator</b>:</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"> <li>◦ Shows success message: "Converted to Fcom code AF1234"</li> <li>◦ Automatically opens betslip with selections loaded.</li> </ul>	<p>@Gary Chen</p> <p>UI should show clear messages, e.g.:</p> <ul style="list-style-type: none"> <li>• "Code not found on [BookieName]. Please check and try again."</li> <li>• "We could only convert 3 out of 5 selections. Missing events: ..."</li> </ul>
Twitter Tag Bot	<ul style="list-style-type: none"> <li>• Tag Fcom official account</li> </ul>	<ul style="list-style-type: none"> <li>• User tag  <a href="#">Football.com - Live Scores &amp; Stats</a> official account</li> <li>• Official account get notified, find the post on twitter by ID</li> <li>• converter the code</li> <li>• reply under comment</li> </ul>	on Twitter
OCR2Code – in-app (share image)	<ul style="list-style-type: none"> <li>• User taps OS share on a screenshot.</li> <li>• Chooses Fcom app from share options.</li> </ul>	<p>1. User sees bet screenshot on social / chat → taps <b>Share</b>.</p> <p>2. In system share sheet, user selects <b>Fcom</b>.</p>	<ul style="list-style-type: none"> <li>• User should be able to <b>leave the screen</b>, continue other things.</li> <li>• When result is ready: <ul style="list-style-type: none"> <li>◦ Notification / in-app message</li> </ul> </li> </ul>

- |  |  |  |
|--|--|--|
|  | <p>3. Fcom app opens a “<b>Code Converter</b>” <b>chat UI</b> (or dedicated screen) showing:</p> <ul style="list-style-type: none"> <li>◦ The shared image</li> <li>◦ A message “We’re converting your screenshot into a booking code...”</li> </ul> <p>4. App sends request to backend:</p> <p>5. Backend pushes job to OCR/AI queue:</p> <ol style="list-style-type: none"> <li>a. OCR extracts text → events / markets / odds.</li> <li>b. Mapping layer resolves them to Fcom offers.</li> <li>c. Booking code generated.</li> </ol> <p>6. Once done, backend sends result:</p> <ul style="list-style-type: none"> <li>◦ SUCCESS: Fcom booking code + selection list</li> <li>◦ PARTIAL: some selections missing</li> <li>◦ FAILED: cannot parse / no matches</li> </ul> <p>7. App:</p> <ul style="list-style-type: none"> <li>◦ In chat UI, shows a message: <ul style="list-style-type: none"> <li>▪ “Here’s your converted code: AF1234. Tap to bet.”</li> </ul> </li> <li>◦ When user taps, open betslip with selections.</li> </ul> | <ul style="list-style-type: none"> <li>◦ Chat history retains input image + output code</li> </ul> |
|--|--|--|

OCR2Code – Telegram bot	<ol style="list-style-type: none"> <li>1. User forwards or uploads a bet screenshot to <b>Fcom TG Bot</b>.</li> <li>2. Bot receives image and sends to backend with:</li> <li>3. Backend runs same OCR + mapping + booking code generator.</li> <li>4. Bot replies in chat: <ul style="list-style-type: none"> <li>◦ On success: <ul style="list-style-type: none"> <li>▪ “Converted booking code: AF1234”</li> <li>▪ Optionally list selections.</li> </ul> </li> <li>◦ On partial/failed: explain what went wrong.</li> </ul> </li> </ol>		
Prompt2Code – app widget or Telegram (Phase 2)	<ol style="list-style-type: none"> <li>1. User opens Code Converter chat (app) or TG Bot and types a prompt, e.g.: <ol style="list-style-type: none"> <li>a. “Find 3 matches starting today with the highest win probability.”</li> </ol> </li> <li>2. Client sends request</li> <li>3. Backend: <ol style="list-style-type: none"> <li>a. Uses AI model to parse intent &amp; constraints: <ul style="list-style-type: none"> <li>▪ number of events: 3</li> <li>▪ date: today</li> <li>▪ criteria: highest win probability</li> </ul> </li> <li>b. Queries Fcom DB / recommendation</li> </ol> </li> </ol>		

	<p>engine for suitable selections.</p> <p>c. Constructs booking code.</p> <p>4. Returns result → booking code + explanation.</p> <p>5. UI shows:</p> <ul style="list-style-type: none"> <li>◦ “Here is a code based on your request: AF5678”</li> <li>◦ List events &amp; markets.</li> </ul>	
--	---	--

Rank	Bookies	Country	Author						
1	<b>Sporty</b>	NG, GH							
2	MSport	NG/GH/UG/ZM	@Danny Lin						
3	Bet9ja	--	@Gary Chen						
4	Bangbet	KE, UG, NG, GH, TZ	@Danny Lin						
5.	BetKing	NG	@Gary Chen						
6	1xBet	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>1xbet - Nigeria</td> <td>1xbet - Global</td> <td>1xbet - Cameroon</td> </tr> <tr> <td>1xbet - Ghana</td> <td>1xbet - Kenya</td> <td>1xbet - Uganda</td> </tr> </table>	1xbet - Nigeria	1xbet - Global	1xbet - Cameroon	1xbet - Ghana	1xbet - Kenya	1xbet - Uganda	
1xbet - Nigeria	1xbet - Global	1xbet - Cameroon							
1xbet - Ghana	1xbet - Kenya	1xbet - Uganda							
7	Betway	Global							
8	LivScore Bet	<a href="https://www.livescore.com/betting/">LiveScore Bet UK   Online Sports Betting &amp; Odds Site</a>							
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
Code2Code BE	@Gary Chen @Danny Lin	IN PROGRESS	<input type="checkbox"/> Integrate bookies <input type="checkbox"/> Mapping events/markets & outcomes <input type="checkbox"/> Complete the code conversion mechanism

Code2Code Client tech			<input type="checkbox"/> load code component expand

## 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):**

- **Monitor engagement**

- 3 entrances (widget/empty betsip/code center load code tab)
- observe funnels
  - bookies spinner
  - select bookies
  - input booking code
  - load code

- **Conversion success rate**

2.1 Track conversion volume and conversion success rate **by bookie**, calculated separately

2.1.1 Conversion success rate should **explicitly include and track partial results**

2.2 The goal is to identify the **most popular third-party (other) bookies** and to improve the **conversion success rate for those bookies**

2.3 Compare the proportion of **Fcom / Sporty codes** versus **other bookies' codes**

## DA requirements

### ☒ GA, FS events

- Event name: code\_converter\_\_open\_bookie\_spinner

- Parameters
  - key: location
  - widget
  - empty\_betsip
  - code\_center

- Trigger: open bookie spinner

- Event name: code\_converter\_\_choose\_bookies

- Parameters
  - key: bookie
  - key: country
  - key: location
  - widget
  - empty\_betsip

- code\_center
- Trigger: click submit CTA
- Event name: code\_converter\_load\_code
  - Parameters
    - key: bookie
    - key: country
    - key: location
    - widget
    - empty\_betslip
    - code\_center
  - TBD: if this information is available
- Trigger: click load code CTA
- Event name: code\_converter\_load\_code\_successfully
  - Parameters
    - key: bookie
    - key: country
    - key: location
    - widget
    - empty\_betslip
    - code\_center
  - Trigger: get successful response from API

#### BE tables

- Converted booking codes
  - bookie
  - # of total selections
  - # of successful selections
  - TBD: how will the failed selections be stored in the database? what info do we have about the failed selections

#### 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

### [Api endpoint](#)

Page	api endpoint	Memo
Home Page	POST /orders/converter/code	• (New) Convert Code2Code
Home Page	GET /orders/converter/config/provider Countries	• (New) Get provider country config
Home Page	GET /orders/share/:shareCode	• Get order record by share code