



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2025/26



Gruppo 17

Nome: BitByBit

Email: swe.bitbybit@gmail.com

Verbale Riunione Esterna - Numero 4

Stato:	Completato
Versione:	1.0.0
Redattori:	Giovanni Visentin
Verificatori:	Gabriele Scaggiante
Responsabile:	Ferdinando Fracasso
Destinatari:	Miriade Prof. Tullio Vardanega Prof. Cardin

Indice

1	Informazioni generali	3
2	Ordine del Giorno	3
3	Discussioni	3
3.1	Credenziali e Ambiente AWS	3
3.2	Strategia di Deploy e PoC	4
3.3	Architettura del Sistema	4
3.4	Autenticazione e Mobile	4
4	Decisioni	5
5	To Do	5



1. Informazioni generali

Data:	06 febbraio 2026
Ora Inizio:	16:30
Durata:	1h 10m
Luogo:	Google Meet
Presenti:	Manisi Riccardo Scaggiante Gabriele Visentin Giovanni
Miriade:	Annalisa Egidì Arianna Bellino Emanuele Righetto
Assenti:	Fracasso Ferdinando Sanguin Marco Parolin Dennis

2. Ordine del Giorno

- Definizione **Architettura^G** e infrastruttura AWS.
- Chiarimenti su credenziali e permessi (Certificazioni ISO).
- Pianificazione del **PoC^G** e deploy.
- Dettagli tecnici su autenticazione, servizi AI e sviluppo mobile.

3. Discussioni

3.1. Credenziali e Ambiente AWS

L'azienda ha comunicato che, dovendo ottemperare a diverse certificazioni ISO, non è possibile fornire accesso diretto come amministratore all'account AWS principale. È stato concordato l'utilizzo di un account interno con permessi specifici:

- Accesso admin garantito per i servizi: Cognito, Lambda ed EventBridge.



- Non verrà fornito accesso admin per Bedrock; le chiamate API saranno gestite come servizio.

3.2. Strategia di Deploy e PoC

Si è discusso il flusso di lavoro per lo sviluppo. L'idea approvata è di realizzare inizialmente il Proof of Concept (**PoC^G**) in locale. Una volta validato il **PoC^G**, si procederà con la migrazione sull'account AWS e la generazione dell'APK definitivo.

3.3. Architettura del Sistema

È stata definita la catena di servizi AWS per supportare l'applicativo:

- **Frontend:** Ospitato su bucket S3 e distribuito tramite CloudFront.
- **Backend:** API Gateway gestirà le richieste (es. GET) connettendosi alle Lambda. Le Lambda conterranno la logica applicativa, ricevendo input JSON e interagendo con Amazon DynamoDB per i dati.
- **Intelligenza Artificiale:** Verrà utilizzato Amazon Bedrock tramite chiamate API. In fase di test locale o per il piano free, è possibile utilizzare modelli alternativi (es. Llama) deployati localmente.
- **Email (SES):** Il servizio SES verrà organizzato in topic a cui collegare le email; l'azienda verificherà internamente la gestione specifica.

3.4. Autenticazione e Mobile

- **Cognito:** Gestirà l'autenticazione tramite User Pool, permettendo permessi differenziati. Sarà integrato il login con Google: il frontend comunicherà con il backend che creerà l'**Utente^G** su Cognito.
- **Mobile App:** L'APK verrà generato tramite Flutter. Si è notato che, anche in debug, l'app potrebbe richiedere permessi di sistema specifici all'**Utente^G**.



4. Decisioni

Descrizione decisione	Codice decisione
Sviluppare il PoC^G in locale prima di migrare sull'infrastruttura AWS aziendale	VE RTB 4.1
Adottare l' Architettura^G S3 + CloudFront (Frontend)	VE RTB 4.2
Utilizzare Amazon Cognito integrato con Google per la gestione utenti	VE RTB 4.3
Utilizzare Amazon Bedrock per i servizi AI (o Llama per test locali)	VE RTB 4.4
Adottare API Gateway + Lambda + DynamoDB (Backend)	VE RTB 4.5

5. To Do

Task	Codice decisione	N° Issue ^G Github
Setup ambiente di sviluppo PoC^G (AWS)	VE RTB 4.1	#338
Setup per Frontend	VE RTB 4.2	#344
Modellare autenticazione con Google	VE RTB 4.3	#343
Modellare chat con chatbot utilizzando Amazon Bedrock	VE RTB 4.4	#342
Modellare inserimento immagine in una nota	VE RTB 4.5	#339
Modellare invio mail a contatti fidati	VE RTB 4.5	#340

Firma aziendale:

(Spazio riservato all'azienda per apporre firma o timbro)