PROGETTO DI PREPARAZIONE ALL’ESAME

Creare il branch “bankapp” sul quale scrivere l’applicazione.

Fare un commit per ogni funzionalità.

Scrive un’applicazione bancaria che implementa le seguenti funzionalità:

* Creazione di un Conto Corrente /creacc
* Possibilità di inserire denaro /deposito
* Possibilità di fare un bonifico /creabon
* Possibilità di prelevare denaro /prelievo
* Possibilità di generare un bollettino da pagare con codice univoco /bollettino

La cifra massima depositabile per volta è di 500 €, il prelievo massimo è di 20 €, mentre sul conto Corrente non può avere saldo negativo e non può superare i 5000 €.

Esistono due tipi di bollettini:

* PAGO ORA (identificabili con codice PGOOA)
  + Importo minimo di 5 €, importo massimo di 17 €
* MAV (identificabili con codice MAV)
  + Importo fisso di 16 €

Creare un branch “database”.

Creare un semplice sistema di autenticazione che richiede NOME UTENTE e PASSWORD usando MongoDB.

Il database conterrà una tabella composta così:

* ID UNIVOCO
* NOME UTENTE
* PASSWORD
* SALDO
* BOLLETTINI

I BOLLETTINI CONNESSI saranno un testo JSON così composto:

{

{

id: “id univoco”,

id\_creditore: “proprietario”,

tipo: “PGOOA o MAV”,

importo: “importo”,

pagato: “yes o not”

}

}

Raggruppare tutti i commit di “bankapp” in uno univoco.

Creare un branch “alphatest” e fare il merge dei branch “database” e “bankapp” su di esso.

Fare il deploy su heroku.

TESTING

Raggiungere il 100 % di BRANCH COVERAGE e STATEMENT COVERAGE con JEST sul branch “alphatest”.

API

Creare un branch “api”.

Scrivere delle API che implementano le seguenti funzionalità:

* GET del saldo in € o in $
  + Ottenendo il tasso di cambio da [https://www.alphavantage.co/](https://www.alphavantage.co)
  + B3GGEIAGNOCSWB93
* GET dei bollettini pagati dall’utente
* POST per la creazione di un nuovo utente
* POST per l’aggiunta di bollettino
* PUT per deposito di denaro
* PUT per prelievo di denaro
* DELETE per eliminare il conto (solo se si sono pagati i bollettini)

Ogni chiamata API deve essere documentata su APIARY, e sarà possibile chiamarle usando un codice univoco: TMP01.

Creare un branch “v.0.0.1” e fare il merge di “alphatest” e “api” su questo branch.