



Test Summary Report

BeeHave

Riferimento	C14_TSR_ver1.0
Versione	1.0
Data	22/01/2023
Destinatario	Prof.ssa Filomena Ferrucci, Prof. Fabio Palomba
Presentato da	C14 Team A.P.Hive
Approvato da	Gianmario Voria, Antonio Della Porta



Revision History

Data	Versione	Descrizione	Autori
21/01/2023	0.1	Prima stesura	LB
22/01/2023	1.0	Revisione per consegna	Tutto il team

Team Members

Nome	Ruolo	Acronimo	Contatto
Gianmario Voria	Project Manager	GV	g.voria6@studenti.unisa.it
Antonio Della Porta	Project Manager	ADP	a.dellaporta26@studenti.unisa.it †
Luigi Bacco	Team Member	LB	l.bacco2@studenti.unisa.it
Irene Gaita	Team Member	IG	i.gaita1@studenti.unisa.it
Maria Lucia Fede	Team Member	MLF	m.fede1@studenti.unisa.it
Gianluca Ronga	Team Member	GR	g.ronga5@studenti.unisa.it
Thomas De Palma	Team Member	TDP	t.depalma@studenti.unisa.it



Sommario

Revision History	2
Team Members.....	2
1. Introduzione	3
2. Relazioni con altri documenti	3
3. Testing Unitario	4
4. Testing di sistema	4

1. Introduzione

BeeHave ha come obbiettivo principale quello di sensibilizzare le persone sulla tematica della lotta all'estinzione delle api. Attraverso una piattaforma online viene fornita agli apicoltori la possibilità di autofinanziarsi creando uno spazio virtuale sul quale mettere in adozione i propri alveari e gestire la vendita del miele prodotto.

Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing pianificate per la suddetta piattaforma. Il fine è quello di garantire che ogni aspetto funzioni in modo corretto.

2. Relazioni con altri documenti

Di seguito la relazione di tale documento con gli altri documenti di testing.

Test Plan (TP)

Il Test Summary Report fa riferimento alle attività di testing specificate nel Test Plan.

Test Case Specification (TCS)

Il Test Summary Report contiene il sunto dell'esecuzione dei test di sistema specificati nel Test Case Specification.

Test Incident Report (TIR)

Il Test Summary Report contiene il sunto dei risultati dell'esecuzione specificati nel Test Incident Report.



3. Testing Unitario

I testing di unità sono stati scritti utilizzando la libreria pytest. Essa dà la possibilità di creare i mock necessari all'esecuzione dei test utilizzando le fixture. Una volta effettuati i test e convalidati da pytest, ogni membro del team deve effettuare il push sul branch apposito e creare una pull request, in modo che i PM avessero modo di supervisionare la correttezza dei test e verificarne l'esito. In caso di problemi, lo stesso sviluppatore era incaricato a risolverli.

4. Testing di sistema

Di seguito vengono riportati i risultati delle esecuzioni dei tests.

Esecuzione	Test non passati	Test passati
Esecuzione 1	3	24
Esecuzione 2	0	27