22MD0115 Dairov Olzhas
22MD0197 Sabit Rayan

LAB-3

To pinpoint particular flaws or problems in the amusement park reservation application in lab 3, we conducted black box testing. In the beginning, it was used to monitor how they had adjusted to various stimuli. We are aware of and can see below that as test cases raise, our odds of covering all bags in application grow. To avoid the production of hand-cases, we opt to automate it utilizing activities that result in combinatorial testing and csv files (all files pinned in the message). As soon as we got the astc-output, we started importing all test cases into the Booking application itself using two algorithms: IPOG and IPOG-F. This resulted in several incorrect media reports. As a result, we could see that IPOG-F functioned well at the conclusion after covering more failures up to strength in the beginning. The information below shows all data and the stages at which mistakes occurred.

IPOG

Strength	Algorithm	N of generated test cases	N of failures, including duplicates	Set of unique failures	Number of unique failures	Cumulative number of failures up to Strength
1	IPOG	12	0	0	0	0
2	IPOG	97	0	0	0	0
3	IPOG	522	6	Error1, Error3	Error1=4, Error3=2	2
4	IPOG	2428	33	Error1, Error2	Error1=20, Error2=13	3
5	IPOG	9681	123	Error1, Error2, Error3	Error1=77, Error2=35, Error3=11	3
6	IPOG	29071	394	Error1, Error2, Error3, Error4	Error1=243, Error2=124, Error3=25, Error4=2	4

IPOG-F

Strength	Algorithm	N of generated test cases	N of failures, including duplicates	Set of unique failures	Number of unique failures	Cumulative number of failures up to Strength
1	IPOG-F	12	0	0	0	0
2	IPOG-F	96	1	Error1	Error1=1	1
3	IPOG-F	481	6	Error1, Error2	Error1=4, Error2=2	2
4	IPOG-F	2402	31	Error1, Error2	Error1=20, Error2=11	2
5	IPOG-F	9604	125	Error1, Error2, Error3, Error4	Error1=78, Error2=41, Error3=5, Error4=1	4
6	IPOG-F	29167	391	Error1, Error2, Error3, Error4	Error1=242, Error2=121, Error3=24, Error4=2, Error5=2	5

Points scored