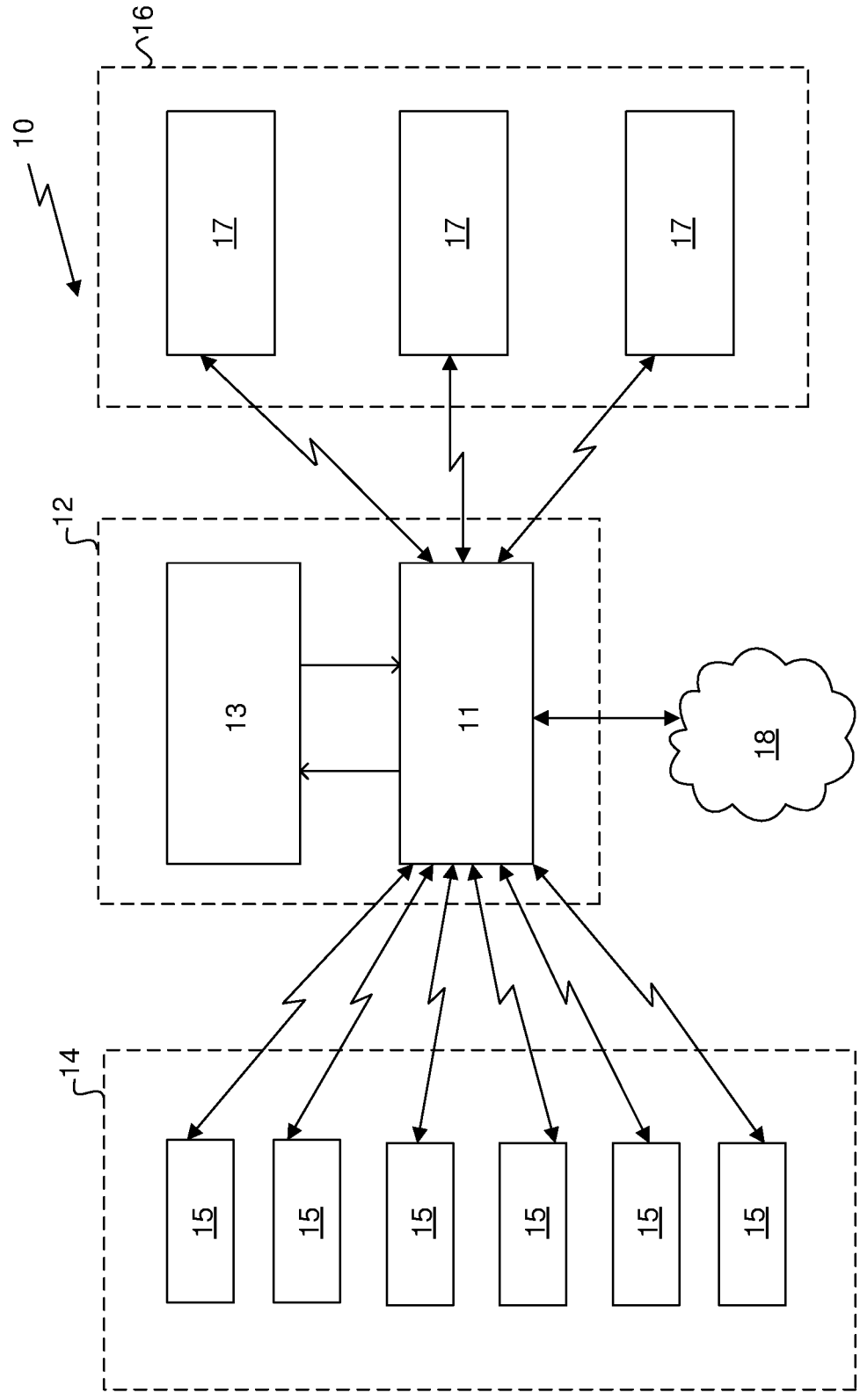


Fig. 1

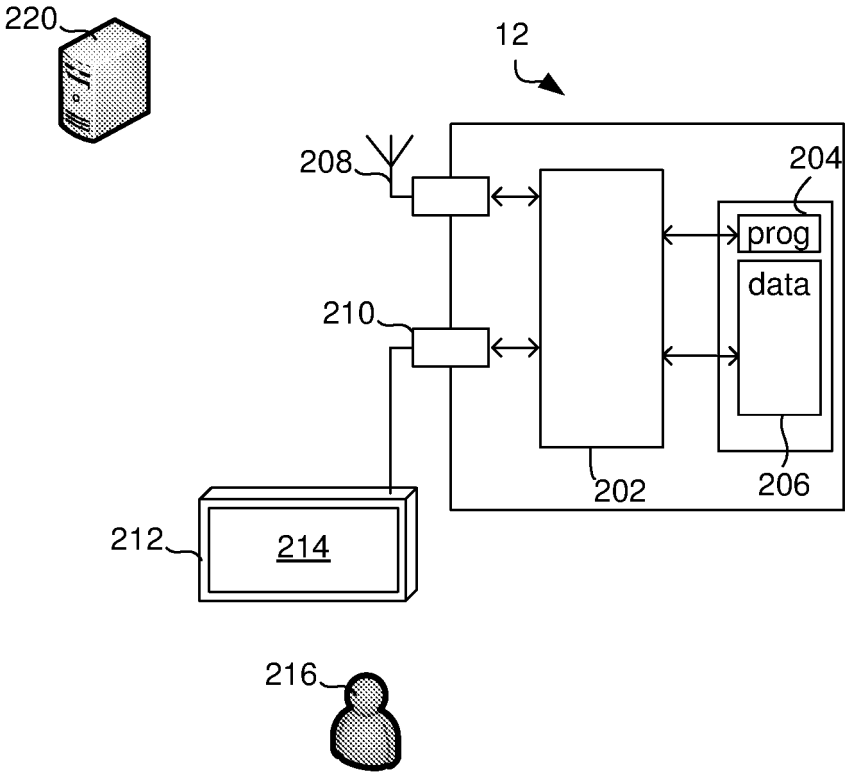


Fig. 2

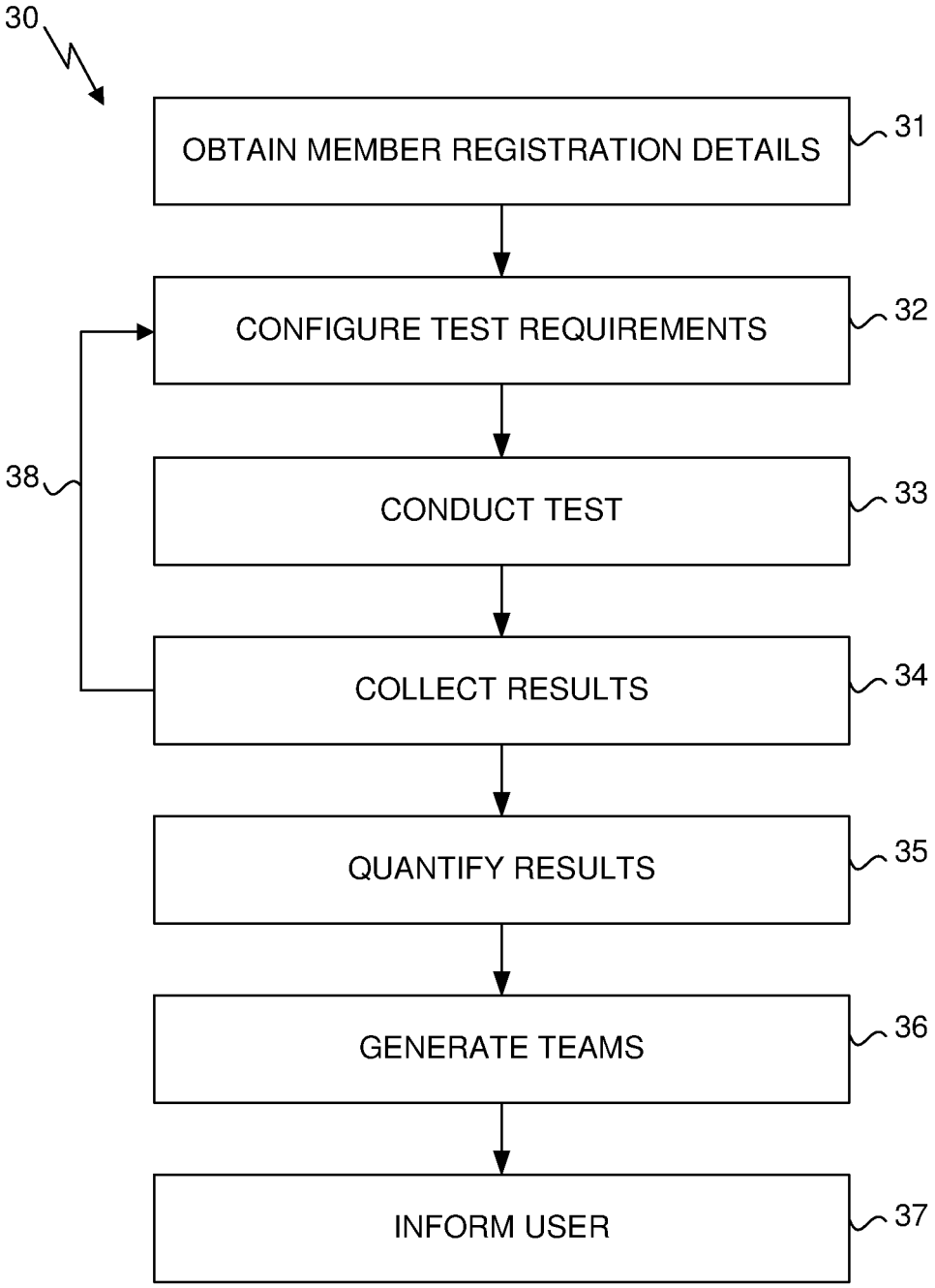

30
OBTAIN MEMBER REGISTRATION DETAILS
31
CONFIGURE TEST REQUIREMENTS
32
38
CONDUCT TEST
33
COLLECT RESULTS
34
QUANTIFY RESULTS
35
GENERATE TEAMS
36
INFORM USER
37

Fig. 3

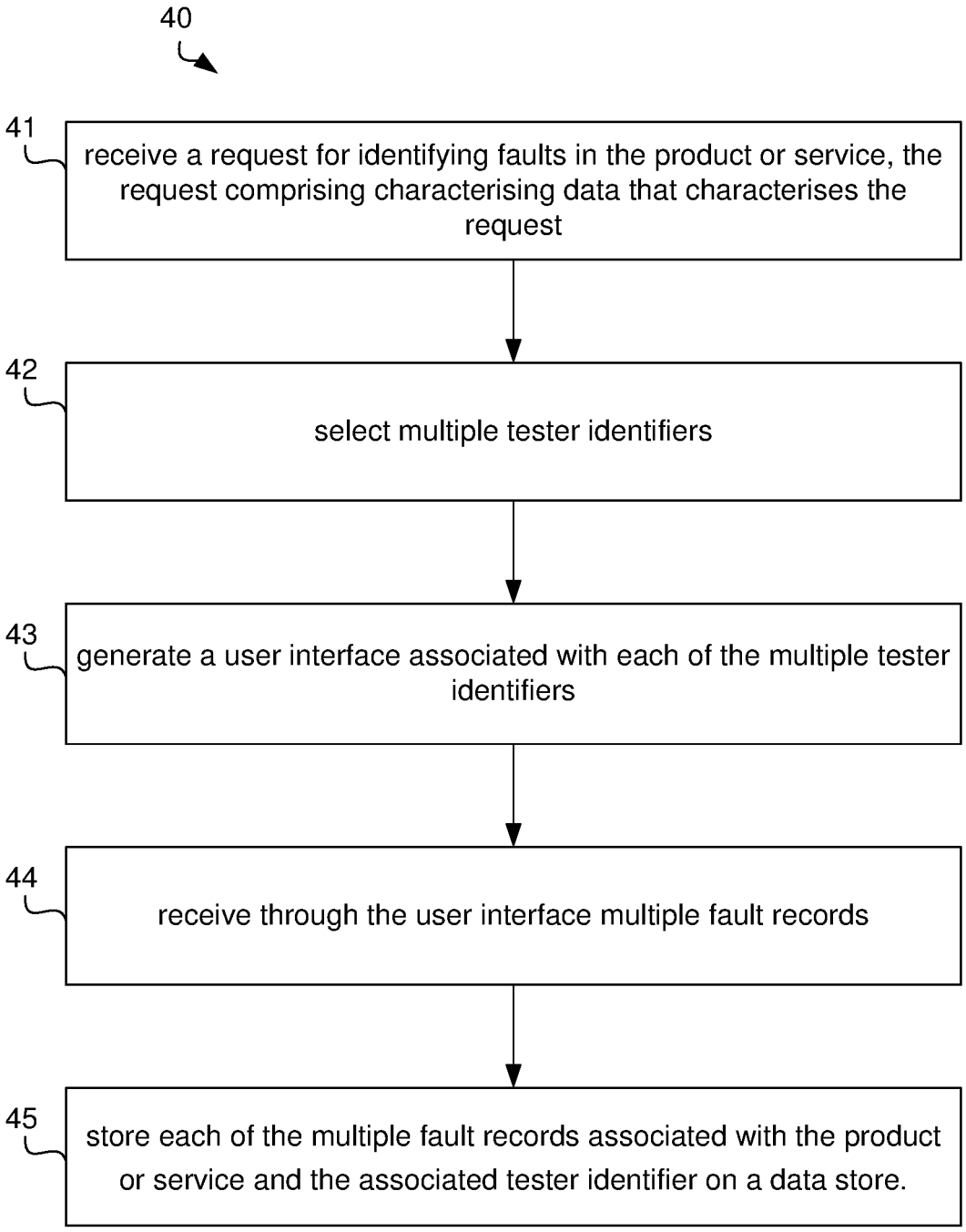

40
41
receive a request for identifying faults in the product or service, the request comprising characterising data that characterises the request
42
select multiple tester identifiers
43
generate a user interface associated with each of the multiple tester identifiers
44
receive through the user interface multiple fault records
45
store each of the multiple fault records associated with the product or service and the associated tester identifier on a data store.

Fig. 4

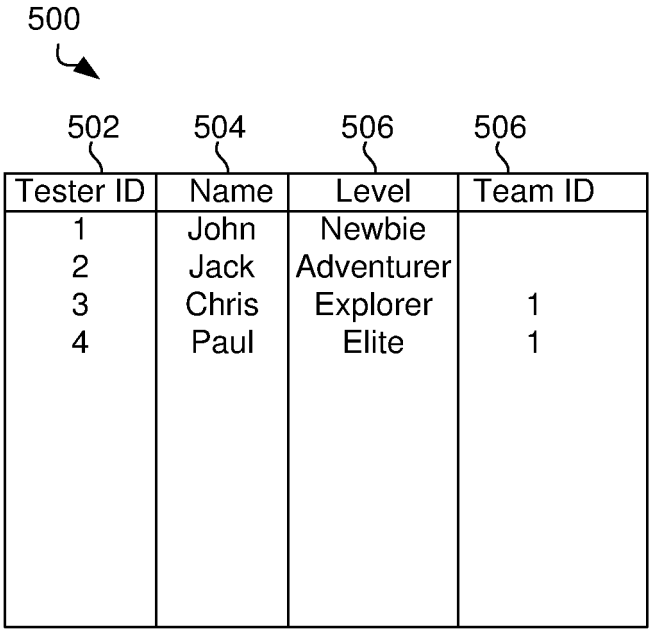

500
502
504
506
506
Tester ID
Name
Level
Team ID
1
John
Newbie
2
Jack
Adventurer
3
Chris
Explorer
1
4
Paul
Elite
1

Fig. 5

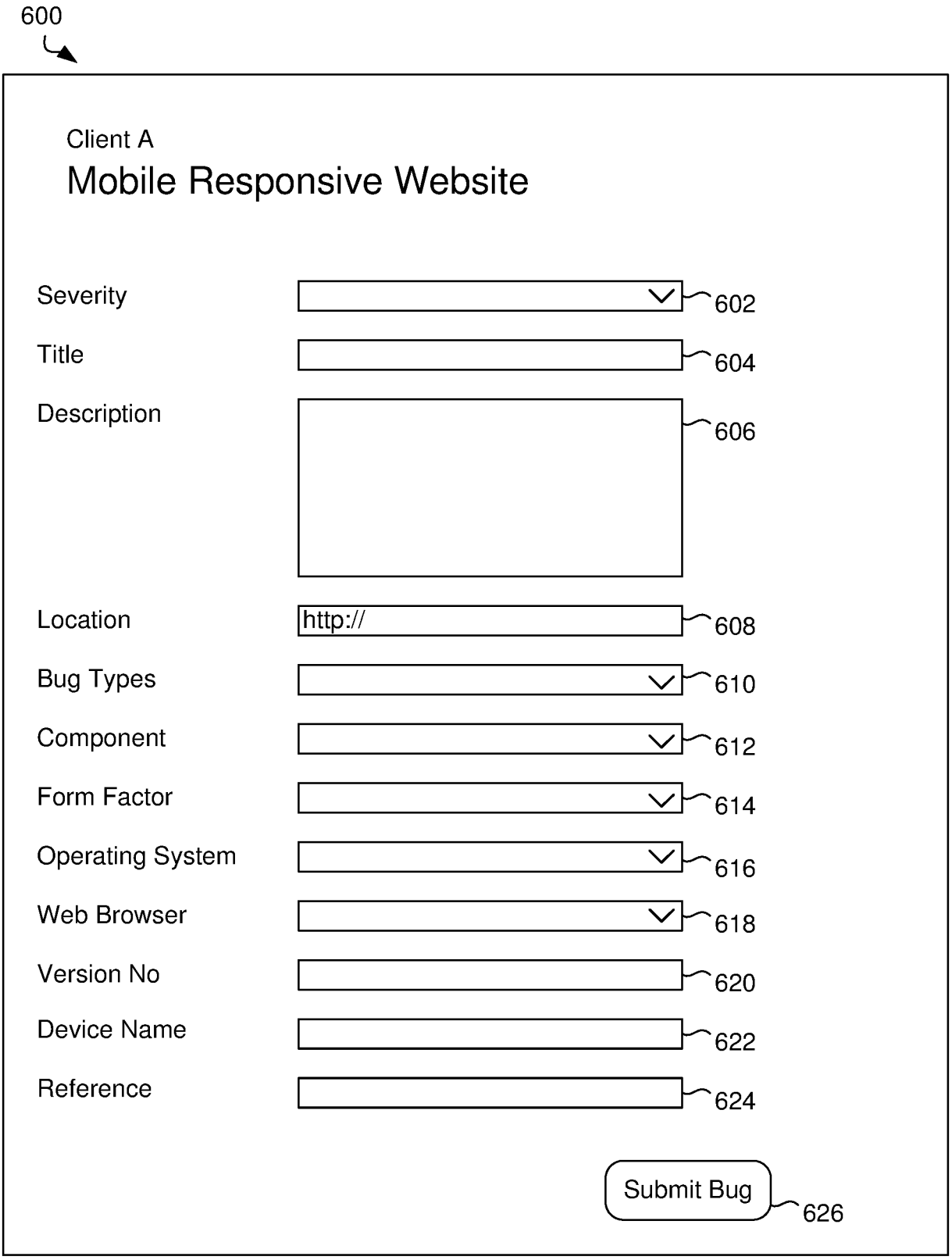

600
Client A
Mobile Responsive Website
Severity
602
Title
604
Description
606
Location
http://
608
Bug Types
610
Component
612
Form Factor
614
Operating System
616
Web Browser
618
Version No
620
Device Name
622
Reference
624
Submit Bug
626

Fig. 6

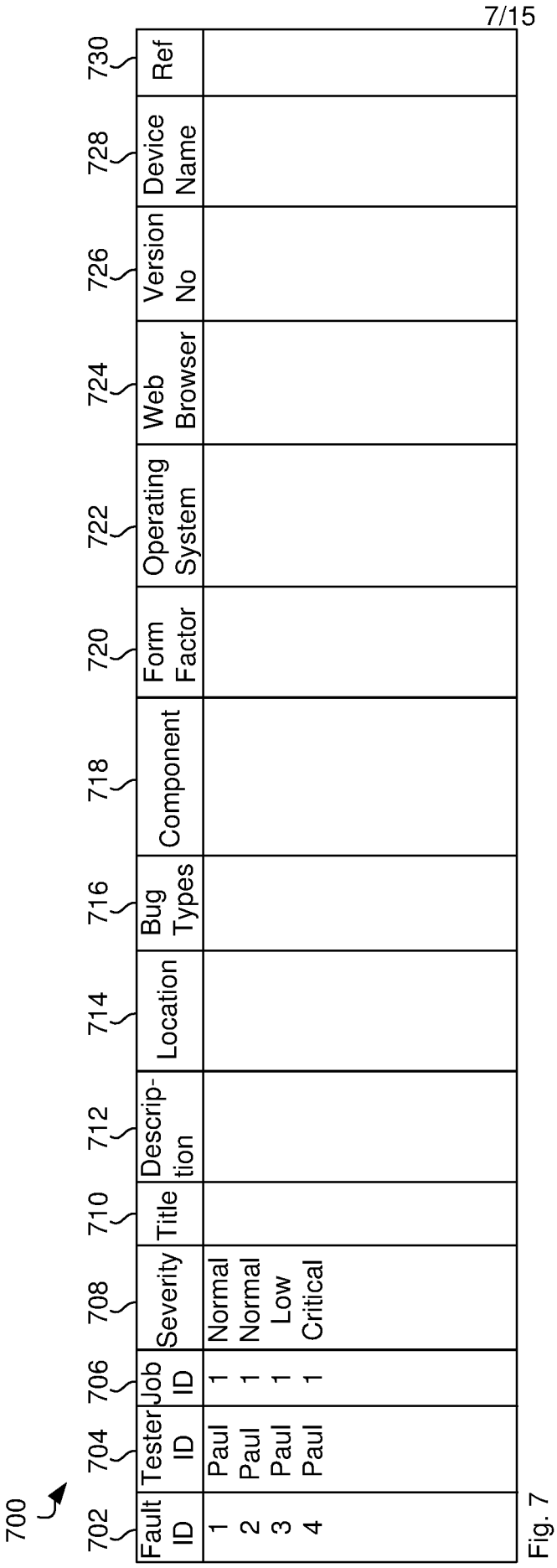

700
702 Fault ID
704 Tester ID
706 Job ID
708 Severity
710 Title
712 Descrip-tion
714 Location
716 Bug Types
718 Component
720 Form Factor
722 Operating System
724 Web Browser
726 Version No
728 Device Name
730 Ref
1 Paul 1 Normal
2 Paul 1 Normal
3 Paul 1 Low
4 Paul 1 Critical

Fig. 7

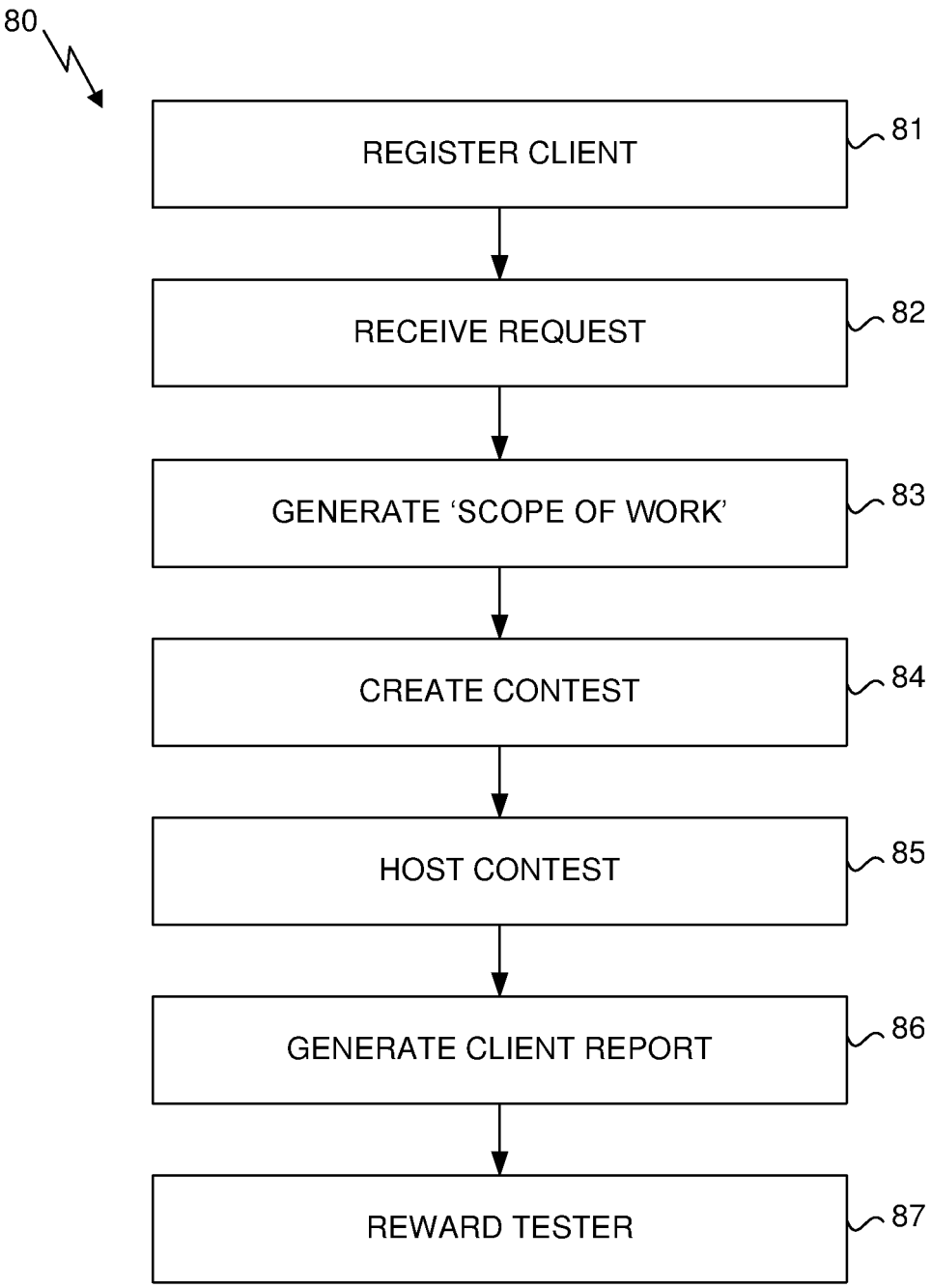

80
REGISTER CLIENT
81
RECEIVE REQUEST
82
GENERATE 'SCOPE OF WORK'
83
CREATE CONTEST
84
HOST CONTEST
85
GENERATE CLIENT REPORT
86
REWARD TESTER
87

Fig. 8

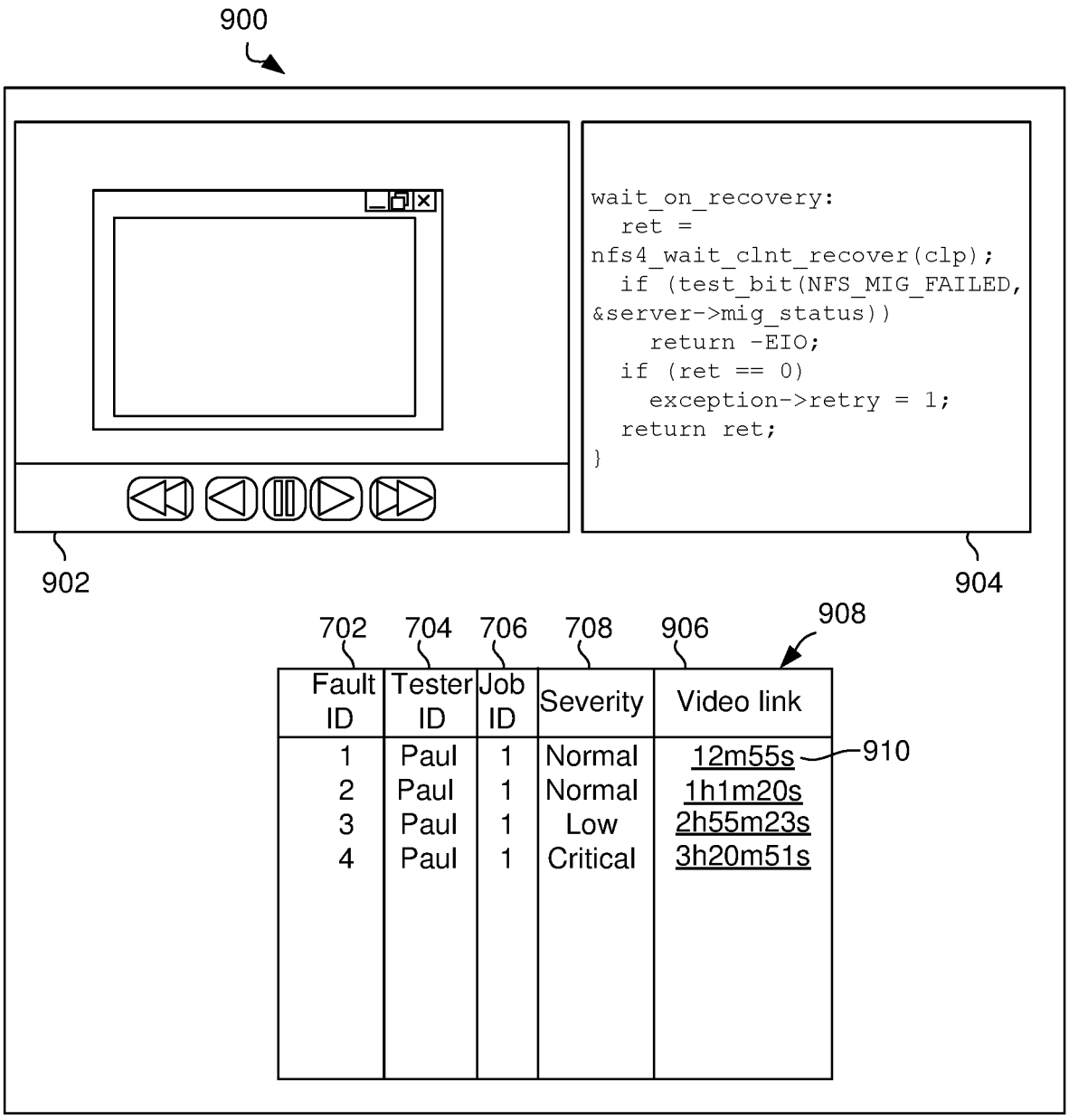

900
wait_on_recovery:
ret =
nfs4_wait_clnt_recover(clp);
if (test_bit(NFS_MIG_FAILED,
&server->mig_status))
return -EIO;
if (ret == 0)
exception->retry = 1;
return ret;
}
902
904
702
704
706
708
906
908
910
Fault ID
Tester ID
Job ID
Severity
Video link
1 Paul 1 Normal 12m55s
2 Paul 1 Normal 1h1m20s
3 Paul 1 Low 2h55m23s
4 Paul 1 Critical 3h20m51s

Fig. 9

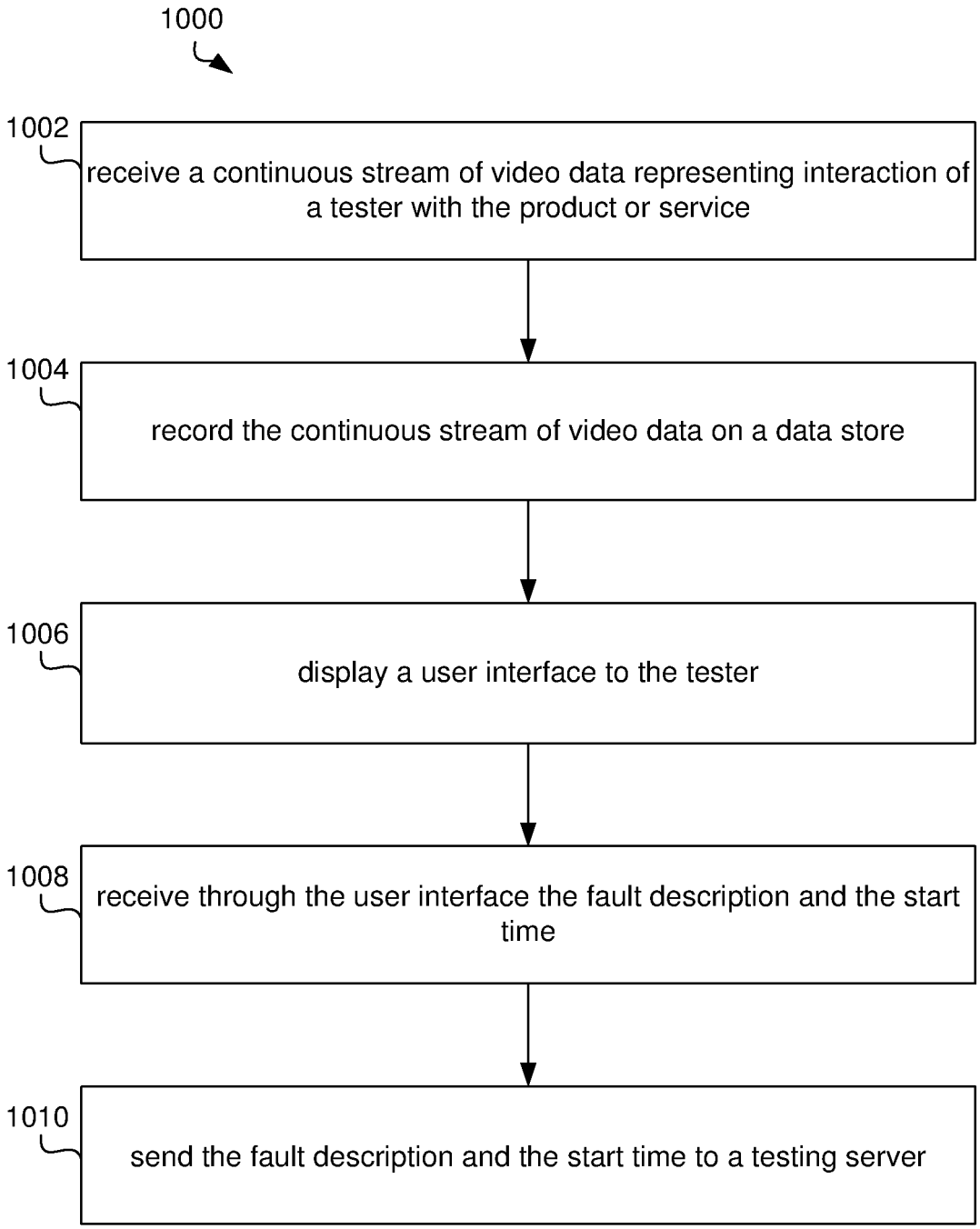

1000
1002
receive a continuous stream of video data representing interaction of a tester with the product or service
1004
record the continuous stream of video data on a data store
1006
display a user interface to the tester
1008
receive through the user interface the fault description and the start time
1010
send the fault description and the start time to a testing server

Fig. 10

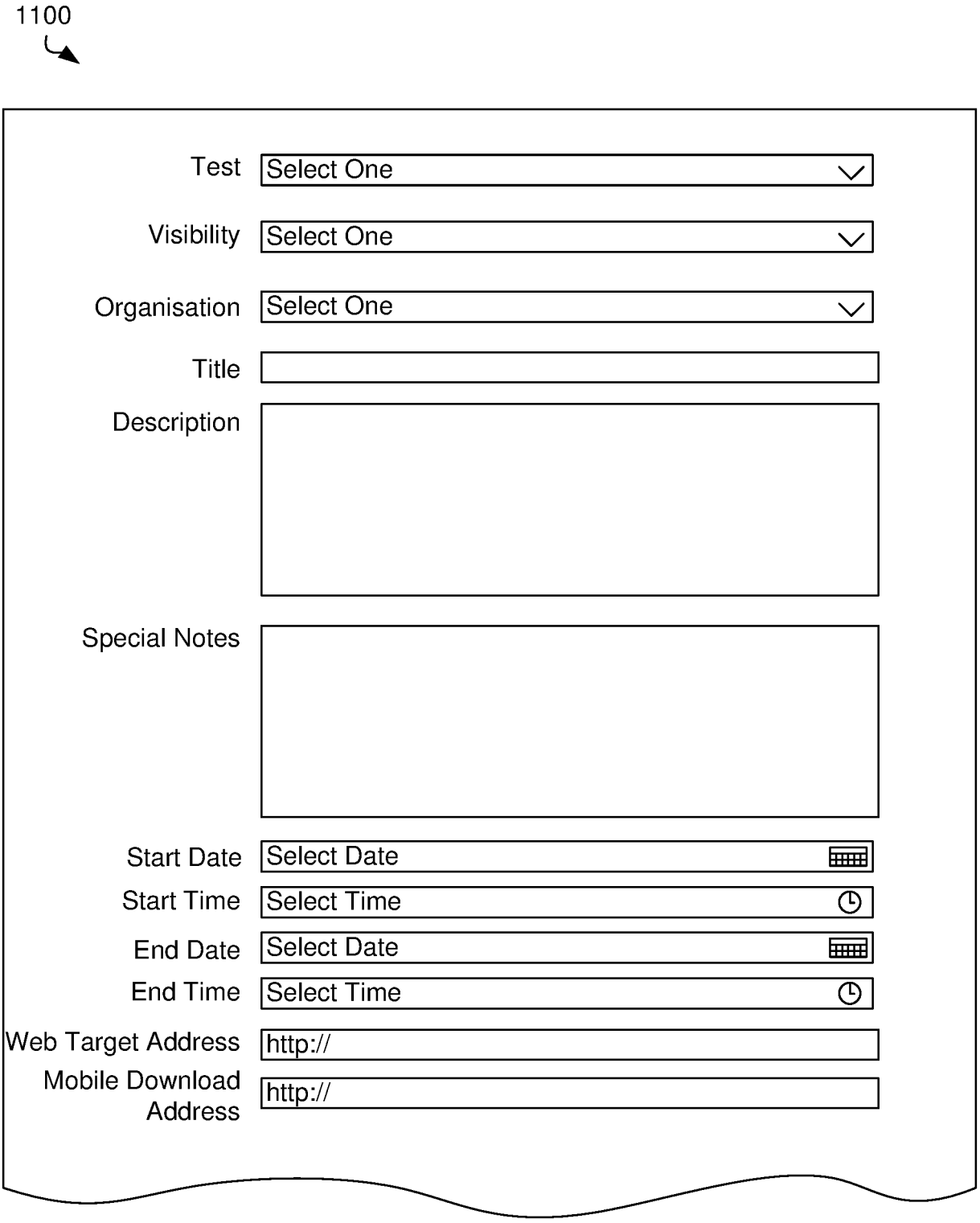

1100
Test
Select One
Visibility
Select One
Organisation
Select One
Title
Description
Special Notes
Start Date
Select Date
Start Time
Select Time
End Date
Select Date
End Time
Select Time
Web Target Address
http://
Mobile Download Address
http://

Fig. 11a

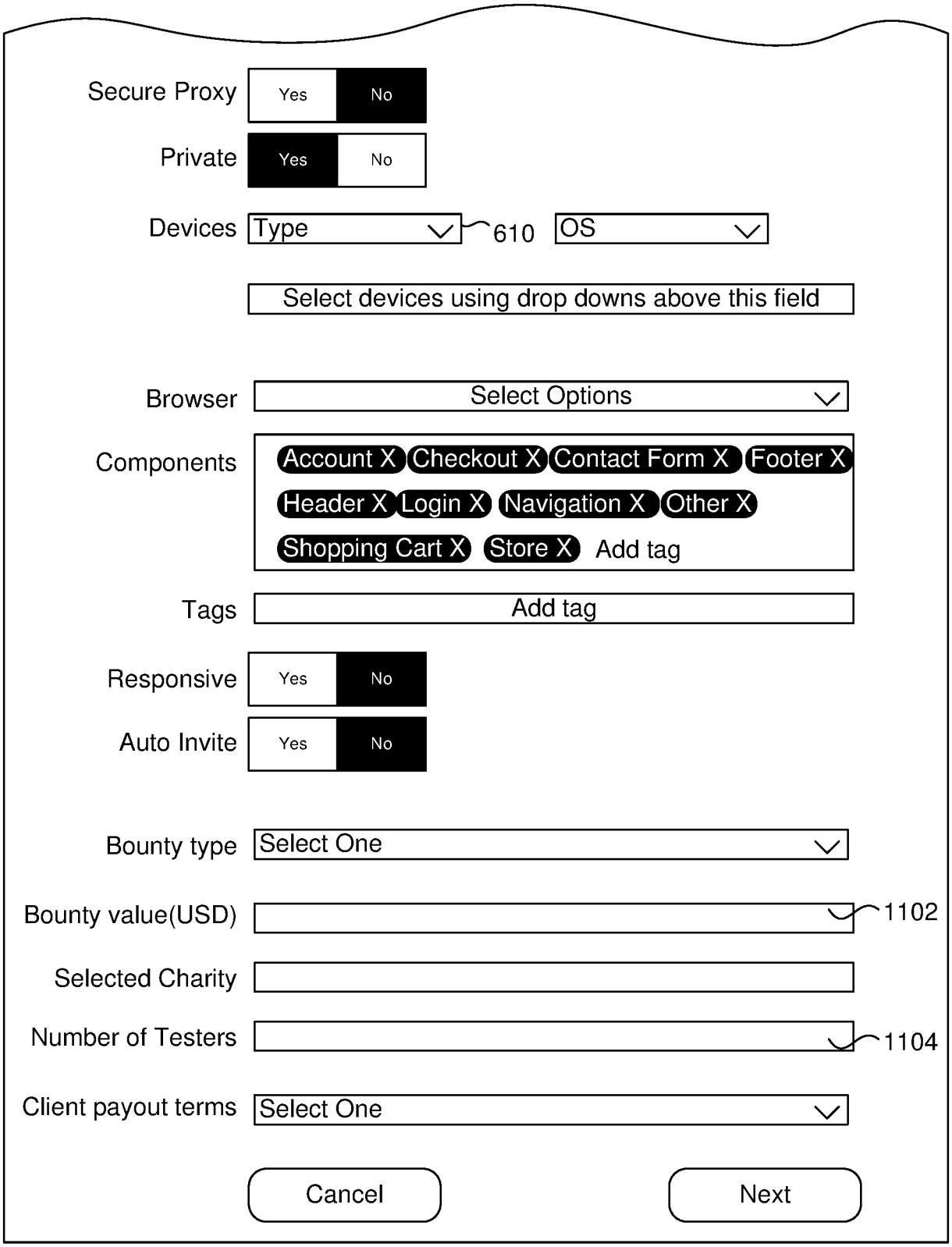

Secure Proxy
Yes
No
Private
Yes
No
Devices
Type
610
OS
Select devices using drop downs above this field
Browser
Select Options
Components
Account X
Checkout X
Contact Form X
Footer X
Header X
Login X
Navigation X
Other X
Shopping Cart X
Store X
Add tag
Tags
Add tag
Responsive
Yes
No
Auto Invite
Yes
No
Bounty type
Select One
Bounty value(USD)
1102
Selected Charity
Number of Testers
1104
Client payout terms
Select One
Cancel
Next

Fig. 11b

1200

Bug Type (1202)	No of Bugs (1206)	Value of Bug (1204)	Sub total (1208)
Critical	24	$2,500	$60,000
High	37	$1,500	$55,500
Medium	76	$750	$57,000
Low	39	$250	$9,750
Total Value of Bugs			$182,250 (1210)
Cost of Test Cycle			$7,883 (1212)
Return on Investment			2312% (1214)

Fig. 12

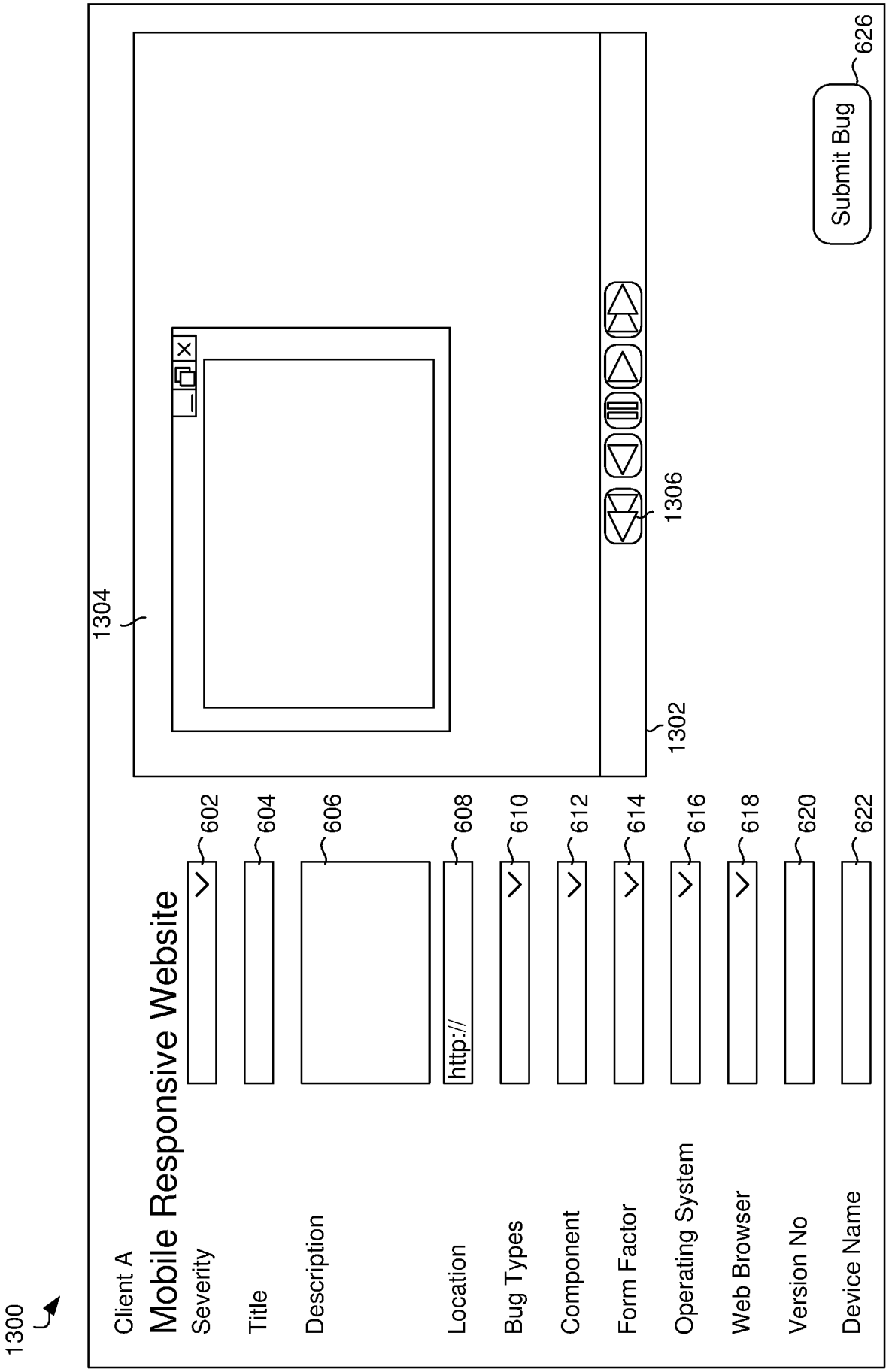

1300
Client A
Mobile Responsive Website
Severity
Title
Description
Location
Bug Types
Component
Form Factor
Operating System
Web Browser
Version No
Device Name
http://
Submit Bug
602
604
606
608
610
612
614
616
618
620
622
626
1302
1304
1306

Fig. 13

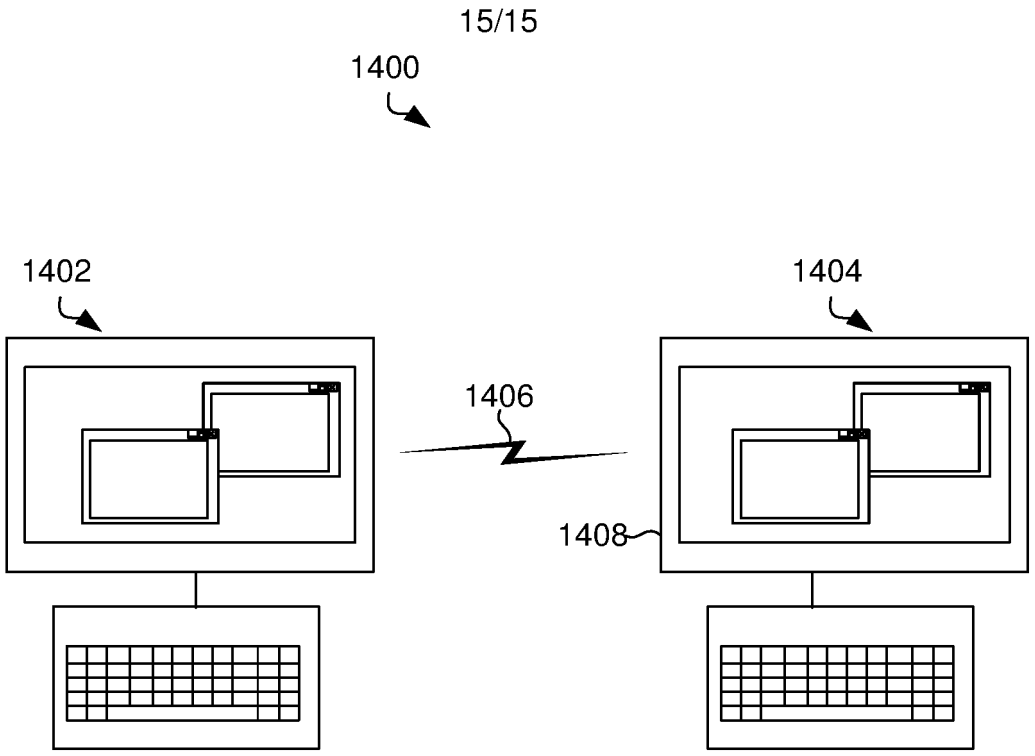


Fig. 14